Contents

INTERNET OF THINGS
(VIII-Sem.)

UNIT 1

IoT DEFINITION, CHARACTERISTICS, IoT CONCEPTUAL AND ARCHITECTURAL FRAMEWORK, COMPONENTS OF IoT ECOSYSTEMS, PHYSICAL AND LOGICAL DESIGN OF IoT, IoT ENABLERS, MODERN DAY IoT APPLICATIONS

Q.1. Write an introductory note on Internet of things (IoT).

Ans. A phenomenon which activates communication between internetworking devices and applications, known as *Internet of things (IoT)*, in which physical objects or things communicate with each other through Internet. Concept of IoT started with object classified as identity communication devices like radio frequency identification device (RFID). These devices are used to tag the objects, for their identification in future. Things are tracked, controlled and monitored by remote computer which is connected through the Internet. The IoT allows the devices for GPS-dependent tracking, controlling and monitoring. It allows for machine-to-machine communication, etc.

It is a reality that the IoT, has helped for creating smart cities. By using the IoT, we will create the self-driving cars functional very soon.

IoT can be defined as the network of physical objects transmitting, receiving or communicating information with the help of Internet and hence enabling the controlling, monitoring and coordinating process across the Internet network.

Q.2. Describe history of IoT.

Ans. The IoT domain leads the world of technology and communication to a new era where objects can communicate, compute and transform the information as per the requirements. This scenario of communication has already been started but didnot get recognition. The term Internet of Things was coined by Kevin Auston, the Executive Director of Auto-ID Labs in MIT in 1999. The concept of IoT first became very popular through the Auto-ID centre in 2003 and in related market analytics and it's publications. When the concept of such communication came into existence, different companies

focused on it and tried to recognize it's significance and began to identify its role and the correlated future aspects, then these companies started investing in the domain of IoT in different periods but at regular intervals of time.

Q.3. Explain the characteristics of IoT.
(R.G.P.V., Nov. 2019)

Ans. The characteristics of IoT are as follows –

(i) *Dynamic and Self-Adapting* – The IoT devices and the systems having the capability to adapt dynamically, the change in contexts and take actions based on their operating conditions, user's context, or sensed environment. For example, a surveillance system comprising of a number of surveillance cameras. These cameras can adapt their modes based on whether it is day or night. They can switch their modes from lower resolution to higher resolution, on the detection of any motion and alert nearby the cameras to do the same.

(ii) *Self-configuring* – A large number of IoT devices can work together to give certain functionality such as weather monitoring because they have self-configuring capability. These devices having ability to configure themselves, setup the networking and fetch latest software upgrades with minimal manual or user intervention.

(iii) *Interoperable Communication Protocols* – A number of interoperable communication protocols are supported by the IoT devices. They can communicate with the other devices and also with the infrastructure.

(iv) *Unique Identity* – Each IoT device has a unique identity and a unique identifier such as an IP address. The IoT systems may have intelligent interfaces which adapt based on the context, allow communicating with users and the environmental contexts. The interface of the IoT device helps users to query the devices, monitor their status, and control them remotely, in association with the control, configuration and management infrastructure.

(v) *Integrated into Information Network* – The IoT devices are integrated into the information network, that permits them to communicate and exchange data with other devices and systems. These devices are used to discovered in the network, by the other devices and or the network having the capability to describe themselves to other devices or user applications. For example, a weather monitoring node can describe its monitoring capabilities to other nodes which are connected to communicate and exchange data.

Q.4. Why do IoT systems have to be self adapting and self configuring ?
(R.G.P.V., May 2018)

Ans. Refer to Q.3 (i) and (ii).

The IoT system is said to be self adapting and self configuring because everyone, from consumers to corporates, is embracing the changes brought by the revolution known as the Internet of things. IoT has changed the world

in many ways than we could have imagined until a few years back. The changes and the advancements will continue in future due to Internet of things, which shape our future. Already the number of sensors are connected with billions of devices and it is estimated that millions of devices will be more interconnected by the next coming years. No price for guessing that the reason this amount is expected to be spent on IoT is that IoT has already shown a lot of potential in a very short time and it has just begun. There are many industries which are impacted by IoT. Some industries had a major influence of IoT while some are just beginning to realise its importance.

Retail companies are investing heavily in IoT because they understand the importance of data-driven analytics and also for the further improvement of the customer experience. On the other hand the customers are enjoying the new experiences made possible due to the IoT. The data driven analytics is based on the data gathered from sensors to reach the potential customers and for better marketing.

IoT is reshaping healthcare as well. Wearable technology to monitor your condition at any time and any place is very common now. Sensors are used for collecting data and at the same time, the data can be visible to the doctors. It helps the doctor to closely monitor the crucial patients from far away. The manufacturing industry is making use of smart machines to improve the overall manufacturing process and to produce the better goods. IoT has something to offer to everyone, it changes the way of doing business, socialize and having fun.

Q.5. Explain about conceptual framework of IoT.

Ans. A general framework contains many devices which share data to a data centre or enterprise or cloud server. IoT framework used in many applications and in enterprise and business processes is therefore more difficult than the equation which describes a simple conceptual framework as given below –

$$IoT = Object + Controller, sensor (physical) and actuators + Internet \quad ...(i)$$

The actions and communication of data at successive levels in IoT having of internetworked devices and objects is shown by the given below equation –

Gather + Enrich + Stream + Manage + Acquire + Organise and analyse
$$= IoT \text{ with connectivity to data centre, enterprise or cloud server} \quad ...(ii)$$

Equation (ii) given by Oracle which is an IoT conceptual framework for the enterprise services and processes. The steps are given below –

(i) Devices data using sensors or the things gather the pre data by the Internet at level 1.

(ii) At level 2, the data enriches by transcoding (i.e. coding or decoding before data sent between two entities) at the gateway.

(iii) At level 3, data streams can be transmitted or received by a communication management subsystem.

(iv) At level 4, data of devices are received by device management, identity management and access management subsystems.
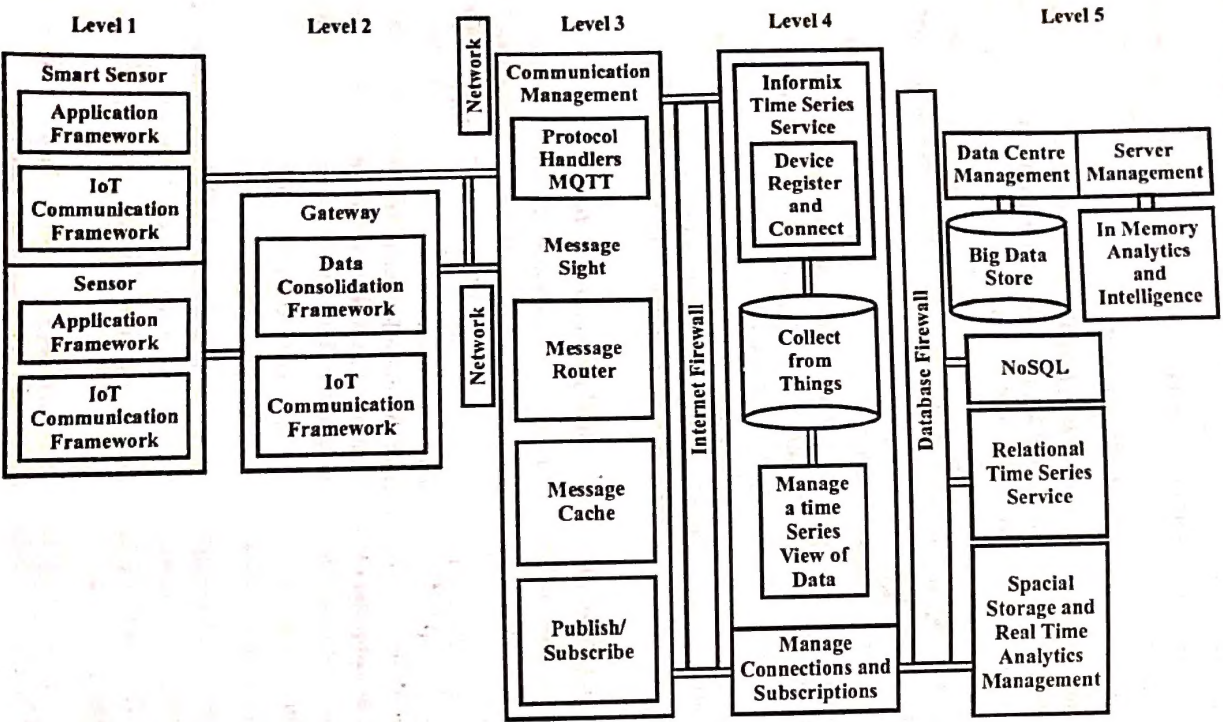
Level 1
- Smart Sensor
  - Application Framework
  - IoT Communication Framework
- Sensor
  - Application Framework
  - IoT Communication Framework

Level 2
- Network
- Gateway
  - Data Consolidation Framework
  - IoT Communication Framework
- Network

Level 3 — *Internet Firewall*
- Communication Management
  - Protocol Handlers MQTT
  - Message Sight
  - Message Router
  - Message Cache
  - Publish/Subscribe

Level 4 — *Database Firewall*
- Informix Time Series Service
  - Device Register and Connect
  - Collect from Things
  - Manage a time Series View of Data
  - Manage Connections and Subscriptions

Level 5
- Data Centre Management
- Server Management
- Big Data Store
- In Memory Analytics and Intelligence
- NoSQL
- Relational Time Series Service
- Spacial Storage and Real Time Analytics Management

Fig. 1.1 IBM Conceptual Framework for IoT

(v) At level 5, data is stored or acquired in database.

(vi) At level 6, data is organised and analysed at this level.

Gather + Consolidate + Connect + Collect + Assemble + Manage and analyse = Internet of things with connectivity to cloud services ...(iii)

A complex conceptual framework is represented by the equation (iii) for IoT using cloud platform based processes and services. The steps are given below –

(i) At level 1, the data is gathered by connecting smart sensor to gateway.

(ii) At level 2, the data is consolidated.

(iii) Between levels 2 and 3, the data streams are communicated by gateway at level 2. At level 3, system employs communication management subsystem.

(iv) At levels 3 and 4, an information service contains collect, connect, assemble and manage subsystems. The services provide from level 4.

(v) At levels 4 and 5, real time series analysis, data analytics and intelligence subsystems are performed.

(vi) At level 5, a data is stored or acquired with a cloud infrastructure.

Q.6. How do the blocks and components in IoT IBM conceptual framework and Oracle reference architectural correlate ? *(R.G.P.V., May 2018)*

Ans. The IoT IBM conceptual framework and Oracle reference architecture are correlated with each other. This can be explained by the following equations which are derived from the IoT IBM conceptual framework and Oracle architecture.

The equation based on the IoT IBM conceptual framework is –

Gather+Consolidate+Connect+Collect+Assemble+Manage and analyse = Internet of things

The equation represent the alternative concept for a complex system. ...(i)

The equation specifies the data communication at successive levels in IoT.

The above equation shows the complex conceptual framework for IoT using cloud-platform based processes and services, it has following steps –

(i) The sensor network is used at level 1 and 2 to gather and consolidate the data. The data of things (devices) using sensors circuits are gathered at level 1. The sensors are connected at the gateway and the data is consolidated at the second level. For example, transformation at the gateway at level 2.

(ii) At level 2, the gateway communicates the data streams between levels 2 and 3. This system uses a communication management subsystem at level 3.

(iii) At levels 3 and 4, an information service consists of connect, collect, assemble and manage subsystems, and the services render from level 4.

(iv) The data analytics, intelligence subsystems and the real time series analysis are also at levels 4 and 5. The cloud infrastructure, data store or database acquires the data at level 5.

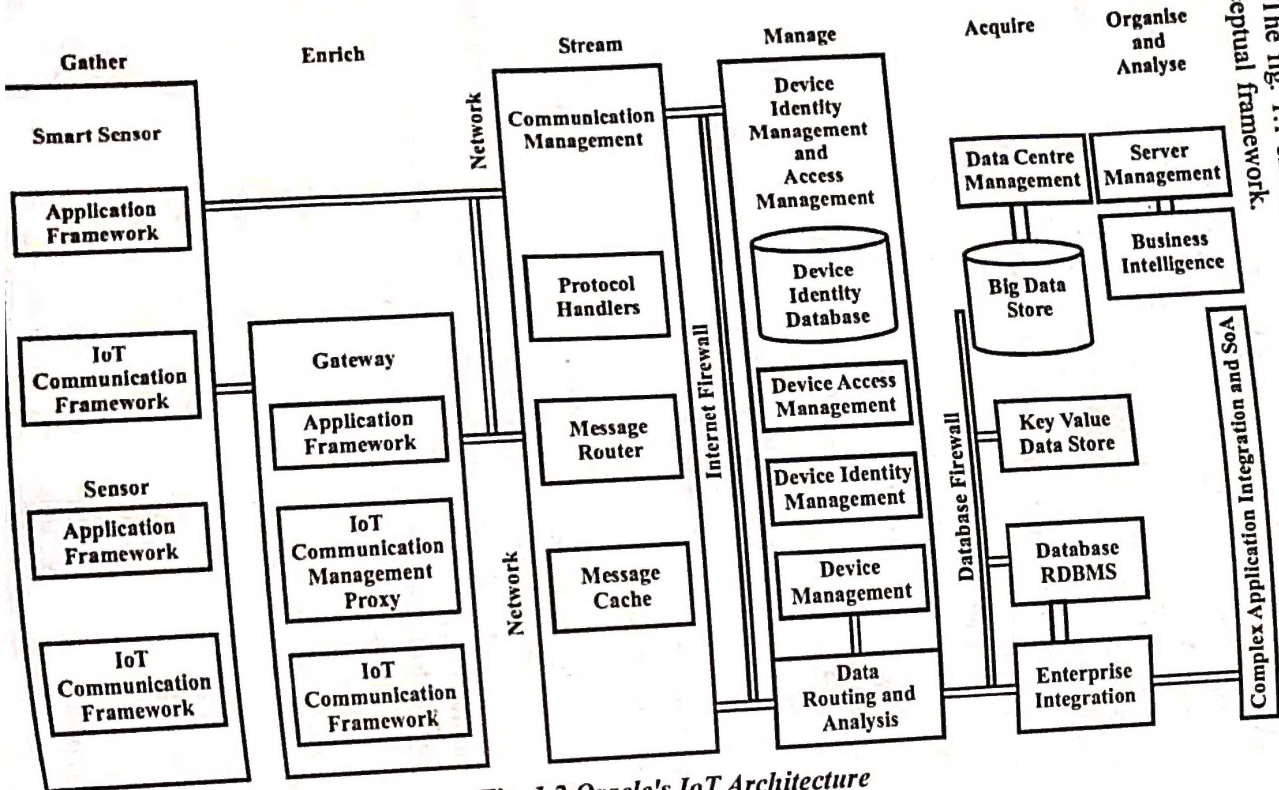The fig. 1.1 shows the blocks and the subsystems for IoT in the IBM conceptual framework.

Fig. 1.2 Oracle's IoT Architecture

The equation based on the IoT conceptual framework of architecture is

Gather + Enrich + Stream + Manage + Acquire + Organise and analyse = Internet of things ...(ii)

The equation (ii) represents the IoT conceptual framework for the enterprise processes and services, based on a IoT architecture suggested by the Oracle, it has following steps –

(i) At first level the data of the devices (things) using sensors or the things gather the pre data from the Internet.

(ii) The sensor connected to a gateway functions as a smart sensor (smart sensor refers to a sensor with computing and communication capacity). The data then enriches at level 2, for example, by transcoding at the gateway, which means coding or decoding before data transfer between two entities.

(iii) The communication management subsystem sends or receives data streams at level 3.

(iv) The access management subsystems, identity management and device management receive the device's data at level 4.

(v) The stored database acquires the data at level 5.

(vi) The routing data from the devices are organised and analysed at level 6. For example, the data analysed for collecting business intelligence in business processes.

The fig. 1.2 shows the Oracle IoT architecture.

Q.7. What are the components of IoT ecosystems ?

Ans. The major components of the IoT ecosystem are IoT devices, device manager, sensor bridge, IoT services and controller. These are discussed below –

(i) IoT Devices – Like other devices, a typical IoT device is also made up of hardware and software. Hardware part consists of sensors, actuators, battery, memory & processing unit (CPU), whereas software part consists of an OS, application software, and preloaded applications. The smart thing in IoT has the responsibility to collect information using different sensors, and the actions are performed using actuators. An example to be considered for this can be a proximity sensor where it can detect an intruder, a thermostat where it receives temperature and humidity conditions of any room and the air conditioner's temperature is set according.

(ii) Device Manager – A device manager acts as a device that coordinates with IoT network and sensor bridge. The primary responsibility of the device manager is to monitor the health and activities of the IoT devices. The reports of the actions and events are provided to the IoT service provider.

(a) IP Network – IP network connects IP enabled IoT devices with device manager. Data can be transferred using TCP/IP or OSI model.

(b) Non-IP Network – Non-IP network connects devices which are enabled to connect with Radio network, Bluetooth, 6LoWPAN, NFC, ZigBee to device manager.

(iii) Sensor Bridge – This component of IoT ecosystem acts as multi-protocol device which receives data from local IoT network process it and forwards it to IoT cloud service or cloud network. The local IoT networks present in the system has a sensor bridge which acts as a connector.

(iv) Data Analytics – Data is processed based on the predefined algorithms and defined criteria and using artificial intelligence, which fulfills our requirements for implementing IoT devices.

(v) IoT Services – The accessing of IoT devices or network can be done using IoT services which are usually hosted on cloud for universal accessibility, this service can be used for device management, service management and based on that any decision can be made.

(vi) Controllers/Users – This component helps in controlling IoT devices deployed in ecosystem.

Q.8. Explain in detail about physical design of IoT.

Ans. Things in IoT means IoT devices, which have unique identities, and can perform remote sensing, monitoring and actuating capabilities. IoT devices may exchange data with other attached devices and applications or transmit the data from other devices and the process the data either locally or transmit the data to centralized servers or cloud-based application back-ends for processing data or perform some operations locally and other tasks within the IoT infrastructure, based on temporal and space constraints.

An IoT device has several interfaces for connections to other devices. These are given below –

(i) I/O interfaces for sensors.
(ii) Memory and storage interfaces.
(iii) Interfaces for Internet connectivity.
(iv) Interfaces for audio/video.

An IoT device can get several kinds of data like temperature, humidity, light intensity from the attached and on-board sensors. The sensed data can be communicated either to other devices or cloud-based storage. IoT devices attach to actuators which permit them to interact with any other object in the vicinity of device.

For Example – An IoT device has a relay switch which is connected to the Internet. When user press on/off through other IoT device, which is also connected to Internet from large distance, the IoT devices on/off according to user command.

IoT Protocols –

(i) Link Layer – Link layer protocols obtain how the data is physically sent over the physical or medium layers of network. The link layer scope is the local network connection to that host can be connected. On the same link, hosts exchange data packets over the link layer using link layer protocols. Link layer finds how the packets can be signaled and coded through the hardware device over the medium to which the host will be connected.

Some important link layer protocols are as given below –

(a) 802.3-Ethernet – It is a group of wired Ethernet standards for the link layer. For example, 802.3 is the standard for 10BASE5 Ethernet which employs coaxial cable for sharing. 802.3.i is the standard for 10BASE-T Ethernet that uses copper twisted-pair connections. 802.3.j is the standard for 10BASE-F Ethernet that uses fiber optic connections. 802.3.j is the standard for 10 Gbit/s Ethernet that uses fiber optic connections. 802.3ae is the standard data rates between 10 Mbit/s and 40 Gbit/s and greater.

(b) 802.11-WiFi – It is a group of wireless local area network communication standards, including extensive description of the link layer.

Example –

(1) 802.11a – It operates in the 5 GHz band
(2) 802.11b – It operates in the 2.4 GHz band
(3) 802.11g – It operates in the 2.4 GHz band
(4) 802.11n – It operates in the 2.4/5 GHz bands
(5) 802.11ac – It operates in the 5 GHz bands
(6) 802.11ad – It operates in the 60 GHz bands

The above standards give 1 Mbit/s to 6.75 Gbit/s data rates.

(c) 802.16-WiMax – IEEE 802.16 is a group of wireless broadband standards. WiMax standards provide 1.5 Mbit/s to 1 Gbit/s data rates. 802.16 m provides 100 Mbit/s data rate for mobile stations and 1 Gbit/s data rate for fixed stations.

(d) 802.15.4-LR-WPAN – IEEE 802.15.4 is a group of low-rate wireless personal area networks. LR-WPAN standards provide 40 kbit/s to 250 kbit/s data rates. The standards give low-cost and low-speed communication for power constrained devices. It form the basic of specifications for high level communication protocol like ZigBee.

(e) **2G/3G/4G-Mobile Communication** – The full form of 2G is second generation. It includes GSM and CDMA. The full form of 3G is third generation. It includes UMTS and CDMA2000. The full form of 4G is fourth generation. It includes LTE and VOLTE. IoT devices based on these standards may communicate over cellular networks. These standards provide data rates from 9.6 kbit/s (for 2G) to 100 Mbit/s (for 4G).

(ii) *Network/Internet Layer* – It is responsible for transmitting IP datagrams from the source host addressing. It performs the packet routing and network to the destination network. IP datagrams contain source and destination IP address for travelling from source to destination through different networks. IP addressing schemes like IPv4 or IPv6 uses for host identification.

(a) **IPv4** – Refer to Q.35.

(b) **IPv6** – Refer to Q.40.

(c) **6LoWPAN** – Refer to Q.10, Unit-III.

(iii) *Transport Layer* – It gives end-to-end message transfer capability independent of the underlying network. The message transfer capability can be set up either using handshakes or without handshakes on connections. The transport layer provides functions like error control, flow control, segmentation and congestion control.

(a) **TCP** – Refer to Q.26, Unit-III.

(b) **UDP** – Refer to Q.27, Unit-III.

(iv) *Application Layer* – How the applications interface with the lower layer protocols to transmit data across network, is defined by the application layer protocols. Application layer protocol encodes application data like files and application data is encapsulated in transport layer protocol which gives connection or transaction oriented communication over the network. For addressing of applications, port numbers are used. Application layer protocols enable process to process connections using ports.

(a) **HTTP** – Full form of HTTP is *hypertext transfer protocol*. It is a application layer protocol which forms the foundation of world wide web. It uses commands like GET, PUT, POST, DELETE, HEAD, TRACE, OPTIONS, etc. Using the HTTP commands, the protocol follows a request-response model in which a client sends requests to a server. Each HTTP request is independent of the other requests. It is a stateless protocol. An HTTP client may be web browser or an application running on the client like application running on client IoT device. It uses universal resource Identifiers to identify HTTP resources.

(b) **CoAP** – Full form of CoAP is constrained application protocol. It is an application layer protocol for machine-to-machine applications.

It uses a client-server model, in which clients communicate with servers using connectionless datagrams.

It is designed to simply interface with HTTP. It uses commands like GET, PUT, POST and DELETE.

(c) **WebSocket** – This protocol permits full-duplex communication over a single socket connection for transmitting messages between client and server. It supports client-server architecture. The client can be a browser or IoT device. It is based on TCP. When TCP connection is open, web socket allows streams of message to be sent back and forth between the client and server.

(d) **MQTT** – The full form of MQTT is message queue telemetry transport. It is a light weight messaging protocol based on the publish-subscribe model. It uses a client-server architecture in which the client like IoT device connects to the server and publishes messages to topics on the server. The broker forwards the messages to the clients subscribed to topics. MQTT is well suited for constrained environments in which the devices have limited processing and resources of memory and low network bandwidth.

(e) **XMPP** – The full form of XMPP is extensible messaging and presence protocol. It supports real-time communication and streaming XML data between network entities. It supports wide range of applications like messaging, presence, data syndication, gaming, multi-party chat and voice or video calls. It permits transmitting small chunks of XML data between networks in real time. It is a decentralized protocol and employs a client server architecture. XMPP allow both client-to server and server to server communication paths. It permits real-time communication between IoT devices.

(f) **DDS** – The full form of DDS is data distribution service. It is a data-centric middleware standard for device-to device or machine to machine communication. It gives quality of service control and configurable reliability. It employs a publish subscribe model in which publishers create topics to that subscribers may subscribe. Publisher is an object responsible for data distribution and the subscriber is responsible for receiving published data.

(g) **AMQP** – The full form of AMQP is advanced message queuing protocol. It is an open application layer protocol for business messaging. It helps point-to-point and publisher or subscriber models, routing and queuing. AMQP brokers receive messages from publishers and route them over connections to consumers. Publishers publish the messages to exchanges, which then distribute message copies to queues. Messages are either delivered by the broker to the consumers which have subscribed to the queues or the consumers can pull the messages from the queues.

Q.9. Explain about logical design of IoT.

Ans. Logical design of an IoT system is related to an abstract representation of the object and processes without going into low-level specifics of the implementation.

(i) IoT Functional Blocks – An IoT system contains a number of functional blocks that give the system capabilities for identification, actuation, sensing, communication and management.

Functional blocks are given below –

(a) Device – An IoT system contains devices which give the facilities of sensing, monitoring, actuation and control functions.

(b) Communication – It manages the communication for the IoT system.

(c) Services – An IoT system employs several kinds of IoT services like device monitoring services, data publishing services, device control services and device discovery services.

(d) Management – This block give several functions to govern the IoT system.

(e) Security – The IoT system is secured by security functional block and by giving functions like authorization, authentication, message and content integrity and data security.

(f) Application – IoT applications gives an interface which users may employ to control and monitor numerous aspects of the IoT system and it also permit users to view the system status and view or analyze processed data.

(ii) IoT Communication Models – IoT communication models are given below –

(a) Request-response – It is a communication model. In this model, client transmits requests to the server and the server responds to the requests. It is a stateless communication model and each request-response pair is independent of others. If the server gets a request, it decides how to respond, fetches the data, retrieves resource representations, prepares the response, and then transmits the response to the client.
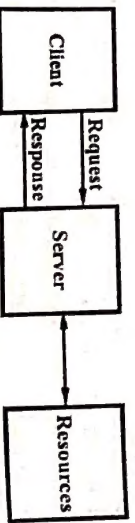

Fig. 1.3 Client-server Interactions in the Request-response Model

(b) Publish-subscribe – It is a communication model which involves publishers, consumers and brokers. Publishers are the data source.

Publishers send the data to the topics. Publishers do not know about consumers. Topics are managed by broker and subscribed by consumers. If the broker receives data for a topic from the publisher, then it sends the data to all the subscribed consumers.

(c) Push-pull – It is a communication model where the data producers push the data to queues and the consumers pull the data from the queues. Producers do not require to know the customer. Queues help in decoupling the messaging between the producers and consumers of data. Queues also behave as buffer that helps in conditions if there is a mismatch between the rate at which the producers push data and consumers pull data.

(d) Exclusive Pair – It is a bidirectional, fully-duplex communication model which employs persistent connection between client and server. It is a stateful communication model. After connection setup, client and server send messages to each other. Server is aware of all the open connections. Once the connection is setup it remains open until the client sends a request to close the connection.

(iii) IoT Communication APIs – IoT communication APIs are given below –

(a) REST-based Communication APIs – The full form of REST is representational state transfer. It is a set of architectural principles. These principles are used for design a web services and web APIs that concentrate on a resources of system and how resource states are addressed and transferred. It follow the request-response communication model. Within a distributed hypermedia system, the REST architectural constraints apply to the components, connectors and data elements.

The REST architectural constraints are given below –

(1) Client-server – The principle behind the client server constraint is the concerns' separation. For example, the server should not be concerned about the user interface, which is a concern of the client. Similarly, the clients should not be concerned with the storage of data which is a concern of the server. Separation permits client and server to be independently developed and updated.

(2) Stateless – Every request must contain all the information necessary to understand the request and cannot take benefit of any stored context on the server in client-server model. Session state is placed entirely on the client.

(3) Cache-able – Cache constraint needs that the data within a response to a request be implicitly or explicitly labeled like cache-able or non cache-able. When a response is cache-able, then a client cache is given

the right to reuse that response data in future for equivalent requests. Caching may partially or fully remove some interactions and improve efficiency and scalability.

(4) Layered System – Layered system constraint requires that the behaviour of components such that every component cannot see beyond the immediate layer with which they are interacting.

(5) Uniform Interface – Uniform interface constraint needs that the communication method between a client and a server should be uniform.

(6) Code on Demand – This constraint is the only one that is optional. Servers may give executable code or scripts for clients to execute in their context.

(b) WebSocket-based Communication APIs – Refer to Q.8(iv)(c).

Q.10. Explain physical and logical design of IoT. *(R.G.P.V., Nov. 2019)*

Ans. Refer to Q.8 and Q.9.

Q.11. Write short note on IoT enablers.

Ans. Internet of Things (IoT) is a global infrastructure for the information society, enabling advanced services by interconnecting (physical and virtual) things based on existing and evolving interoperable information and communication technologies. Enablers of IoT can be grouped into three categories, implementation, connectivity and enabling technologies.

The Internet of Things is not a single technology, but it is a mixture of different hardware and software technology. The Internet of Things provides solutions based on the integration of information technology which includes
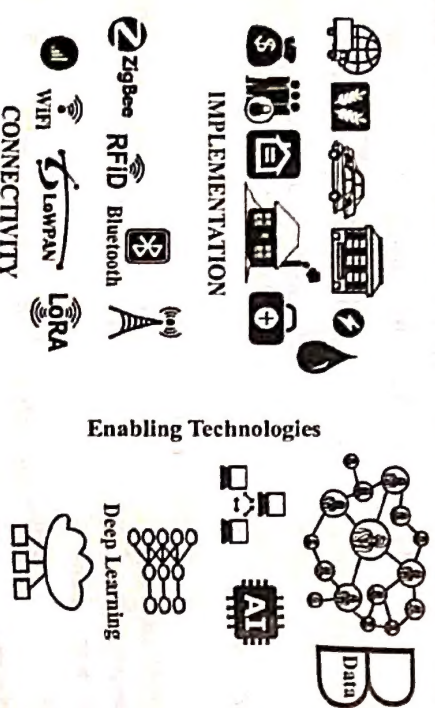

Enabling Technologies

IMPLEMENTATION

ZigBee   RFID   Bluetooth

WiFi   6LowPAN   LORA

CONNECTIVITY

Deep Learning

AI

Data

Fig. 1.4 IoT Enablers

---

electronic systems used for communication between individuals or groups. Examples of standards in these categories include wired and wireless technologies like smart home, banks, transportation sector, agriculture, healthcare, ZigBee, RFID, Wi-Fi, 6LowPAN, LORA, Bluetooth, Deep learning, Bigdata, artificial intelligence sensor network and regular network. The enablers for IoT is shown in fig. 1.4.

Q.12. Write in detail application of Internet of Things.

(R.G.P.V., May 2019)

Or

Write short note on applications of IoT.   *(R.G.P.V., May 2018)*

Ans. An association of platforms, applications, devices and services provides the ability to improve quality of life. The development of a large number of applications will make possible through the great potentialities offered by the IoT. It plays a crucial role in the so-called fourth industrial revolution. The IIC was created with the goal of transforming industry through intelligent, interconnected objects which can improve performance, lower costs and increase reliability. This consortium assumes that industry involves the areas of energy, manufacturing, healthcare, smart cities and transportation. Smart cities, health care, smart homes and buildings, mobility and transportation, energy, industry, agriculture and the environment/planet are the main areas of application. Note that some applications such as environmental monitoring, can fit into different groups like smart city, smart buildings, the environment and industry. It is worth mentioning that in the context of IoT, it is not possible to develop a one-size-fits-all solution. For example, a solution for home environmental monitoring cannot be adequate for industrial ambient monitoring because of several types of physical conditions and relevant parameters. An industrial setting needs several levels of accuracy, security and robustness for the deployed sensors and software, while a home environment generally does not impose such restrictions. Some examples of these applications are given below –

(i) Healthcare – IoT technologies play an important role in healthcare domain, namely, in two areas – clinical care and remote monitoring. IoT technology is used for creating a low-cost, small-sized, low power wearable biosensors, which can improve the quality of life of people suffering from chronic diseases or even during emergencies, either inside or outside their homes. Elderly tracking or ambient assisted living (AAL) encompasses technical systems to support elderly people in their daily routine to allow an independent and safe life-style as long as possible. IoT device will improve the life of disable people. IoT device will use in non-invasive glucose-level sensing for diabetes management.

(ii) Smart Cities – Finding ways to use technology to improve the quality of life in a city has become one of the most popular research topics in the area of IoT applications. For building efficiency, services and safety, smart city solutions include several areas like water and waste management, lighting control, energy, transportation, traffic, and parking management.

(iii) Mobility and Transportation – All types of vehicles in a city are equipped with actuators and sensors, and transported goods are also equipped with sensors and tags. These sensors send important information to traffic control sites like, status of transported goods, better route, creation of innovative solution. Moreover, modern cars are also equipped with several sensors, which provides kinematics information, automotive diagnostic services and so forth. Car can be again equipped with external sensing devices to monitor specific physical parameters, like pollution humidity and temperature. If these data have properly, then it will help to make the road transport greener, smarter and safer. For example, driving recommendations will provide eco-efficiency for public transportation and reduce fuel consumption and emission. When evident through smart vehicle, they immediately call the ambulance. Now mobile applications like Google traffic rely on user-contributed data to monitor traffic conditions. For improving cycling and driving in cities safer and smoother, we will use the concept of smart traffic light infrastructures. Another area of application is modern logistics, which will monitoring the whole process of the physical movement of goods from suppliers to customers.

(iv) Smart Homes and Buildings – IoT devices help in air conditioning control and monitoring and also help in energy saving and even reduce solar green house production. IoT uses for small buildings is necessarily cost savings, providing the building with some intelligence through building automation.

(v) Smart Manufacturing – Manufacturing system requires various types of decision making at various levels of its activities for design and operation of manufacturing system. Hence, IoT technology can be used for developing smart and intelligent manufacturing enterprises. IoT technology in these environment may be employed to serve multiple purposes like environment control, lighting control, safety, to production optimization, error detection and correction, automatic control of stocks.

(vi) Energy – The smart grid is a recent kind of intelligent power system which may improve energy efficiency, minimize environmental impact, improve the safety and reliability of the electricity supply, and minimize the electricity transmission of the grid. The integration of IoT technology in smart grids can help to implement fault detection and monitoring as well as consumption monitoring, by the installation of energy sensors. IoT technology

---

also provide heat and energy management in home and buildings to achieve an energy savings purpose. Using IoT technology to collect data on energy consumption may also help to enhance the energy efficiency and competitiveness of manufacturing companies at the energy production level.

(vii) Smart Planet/Environment – IoT technology can be used for environmental monitoring, meteorological monitoring, pollution control, waste management or disaster monitoring for development and management of cities.

(viii) Smart Agriculture – Modern agriculture contains a various set of requirements compared to traditional agriculture. It should be high quality, high yield, safe, efficient and ecological. IoT technology is use for multiple purposes such as for irrigation control, fertilization, pest control and animal monitoring as well as for greenhouse monitoring, viticulture and horticulture.

Q.13. Discuss area of development and standardization in Internet of things.

Ans. Area of Development – Refer to Q.12.

(R.G.P.V., May 2019)

Standardization is a voluntary cooperation among industry, consumers, public authorities and other interested parties for the development of technical specifications based on consensus. Standardization complements market-based competition, typically in order to achieve objectives such as the interoperability of complementary products/services, to agree on test methods and on requirements for safety, health and environmental performance. Standardization also has a dimension of public interest. Standard makers should be close to standard users/implementers.

Q.14. What do you understand by Internet of Things (IoT) ? Explain any five areas of IoT.

(R.G.P.V., Nov. 2019)

Ans. Refer to Q.1 and Q.12.

Q.15. Enlist the various modern day applications of IoT.

Ans. The various modern day applications of IoT are as follows –

(i) Smart parking
(ii) Structural health
(iii) Noise urban maps
(iv) Smart phone detection
(v) Traffic congestion
(vi) Smart lighting
(vii) Waste management
(viii) Smart roads
(ix) River floods
(x) Smart grid
(xi) Tank level
(xii) Photovoltaic installations
(xiii) Water flow
(xiv) Silos stock calculation
(xv) Perimeter access control
(xvi) Liquid presence

(xvii) Earthquake detection

(xviii) Forest fire detection

(xix) Air pollution

(xx) Snow level monitoring

(xxi) Landslide and avalanche prevention

(xxii) Water leakages

(xxiii) Radiation level

(xxiv) Explosive and hazardous gases

(xxv) Supply chain control

(xxvi) NFC payment

(xxvii) Intelligent shopping applications

(xxviii) Smart product management.

Q.16. What are the advantages and disadvantages of IoT ?

Ans. Advantages of IoT –

(i) Reduced Waste – IoT makes areas of improvement clear. Current analytics give us superficial insight, but IoT provides real-world information leading to more effective management of resources.

(ii) Enhanced Data Collection – Modern data collection suffers from its limitations and its design for passive use. IoT breaks it out of those spaces, and places it exactly where humans really want to go to analyze our world. It allows an accurate picture of everything.

(iii) Improved Customer Engagement – Current analytics suffer from blind-spots and significant flaws in accuracy, and as noted, engagement remains passive. IoT completely transforms this to achieve richer and more effective engagement with audiences.

(iv) Technology Optimization – The same technologies and data which improve the customer experience also improve device use, and aid in more potent improvements to technology. IoT unlocks a world of critical functional and field data.

Disadvantages of IoT –

(i) Security – IoT creates an ecosystem of constantly connected devices communicating over networks. The system offers little control despite many security measures. This leaves users exposed to various kinds of attackers.

(ii) Privacy – The sophistication of IoT provides substantial personal data in extreme detail without the user's active participation.

deployment, and maintenance given their use of multiple technologies and a large set of new enabling technologies.

(iii) Complexity – IoT systems are complicated in terms of design, deployment, and maintenance given their use of multiple technologies and a large set of new enabling technologies.

(iv) Flexibility – Many are concerned about the flexibility of an IoT system to integrate easily with another. They worry about finding themselves with several conflicting or locked systems.

(v) Compliance – IoT, like any other technology in the realm of business, must comply with regulations. Its complexity makes the issue of compliance seem incredibly challenging when many consider standard software compliance a battle.

M2M COMMUNICATIONS, IoT vs M2M, IoT vs WoT, IoT REFERENCE ARCHITECTURE

Q.17. What is machine-to-machine (M2M) ?

Or

Write short note on M2M.   **(R.G.P.V., May 2019)**

Ans. Machine-to-machine is a concept in which two or more machines may communicate with each other, and carry out specific functions. The software and the connectivity on the machines make them semi-intelligent and enable some degree of intelligent functionality if coupled with a certain policy. In IoT, physical object start getting connected to the Internet. That is these "things" were not originally some sort of computing machines but some type of end nodes which can be now enabled to be attached to the Internet.

For example – Lamp can be controlled through smart phone if smart phone is connected to the Internet. Lamp is not computing device yet Internet connectivity and software allows the ability to control them.

Q.18. Draw and explain M2M system architecture.

Ans. For the purpose of remote monitoring and control and data exchange, machine-to-machine is defined as the networking of machines. M2M system architecture is shown in fig. 1.5. It consists of –

(i)   M2M area networks

(ii) Communication network

(iii) Application domain.

(i) M2M Area Networks – It is made up of machines which have embedded hardware modules for communication, actuation and sensing. For M2M local area neworks, several communication protocols like Bluetooth,

ModBus, ZigBee, M-bus, Wireless M-bus, 6LoWPAN, IEEE 802.15.4 power line communication, etc. are used. Connectivity between M2M node is provided by these protocols within an M2M area network. M2M area networks employ either proprietary or non-IP based communication protocols.

(ii) Communication Network – It gives connectivity to remote M2M area networks. The communication network may employ either wired or wireless networks. M2M gateways can be used to enable the communication between remote M2M area networks. For M2M area network, M2M gateway performs protocol translations to enable IP connectivity.

(iii) Application Domain – M2M contains several application domains like home automation, smart grids, smart metering, industrial automation etc.
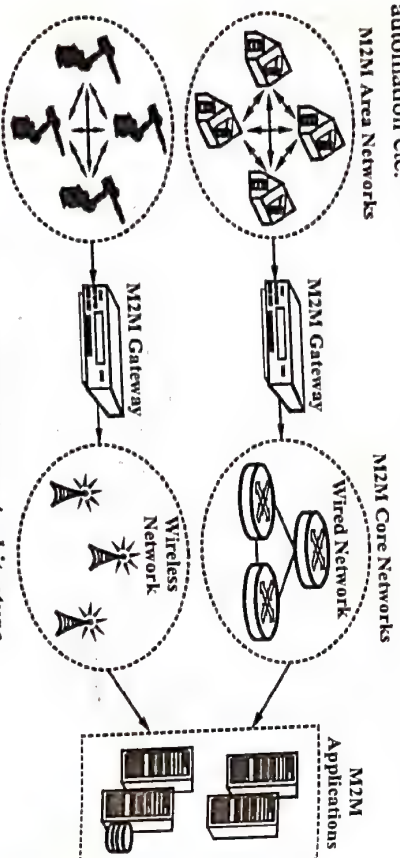
M2M Area Networks

Fig. 1.5 M2M System Architecture

Q.19. Explain in brief M2M towards IoT.

Ans. M2M solution is not new. M2M solutions have been around for decades and are quite common in many different scenarios. Our planet is facing massive challenges like environmental, social and economic. The changes which humanity requires to deal with in the coming decades are unprecedented, not because same things have not happened before during our common history on this planet, at the same time, but because many of them are happening. From constraints on natural resources to a reconfiguration of the economy of worlds, many people thinks, we can solve these problem through technology. Essentially, therefore a set of megatrends are grouping to create needs and capabilities, which in turn produce a set of IoT technology and business drivers. A megatrend is a trend that will have a basic and global impact on society at a macro level over many generations. It is something that will have a significant impact on the world in the foreseeable future.

Q.20. Write down the differences between M2M and IoT.

Ans. The differences between M2M and IoT are described as follows –

(i) Communication Protocols – The communication between the machines or devices happen differently in both M2M and IoT. The communication within the M2M area network is based on the either proprietary or non-IP based protocol. The most commonly used protocols in M2M communication are ZigBee, Bluetooth, ModBus, M-Bus, wireless M-Bus, Power Line Communication (PLC), 6LoWPAN, IEEE 802.15.4, Z-Wave etc. The M2M communication is focused on the protocols below the network layer. Whereas the IoT communication is focused on the protocols above the network layer such as HTTP, CoAP, WebSockets, MQTT, XMPP, DDS, AMQP etc. as shown in fig. 1.6.
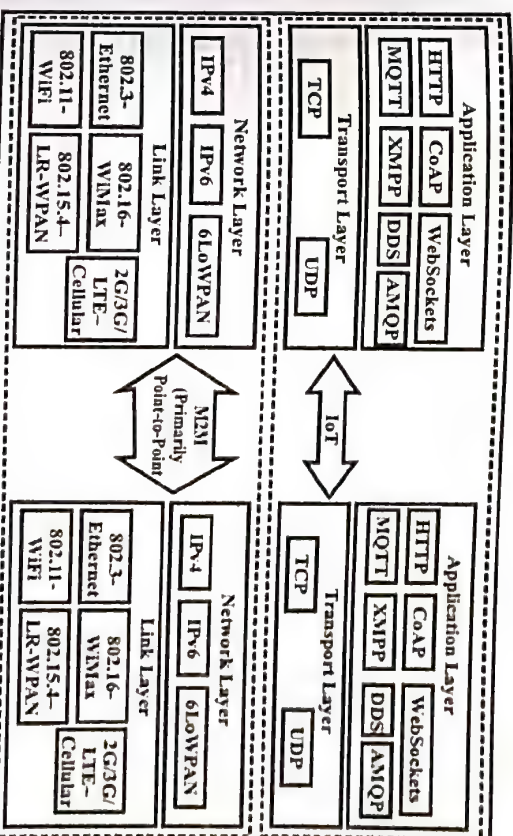
Fig. 1.6 Communication in IoT is IP-based whereas M2M uses non-IP based Networks

(ii) Machines in M2M vs Things in IoT – The physical objects are the 'things' which are referred to the unique identifiers which can sense and communicate with their external environment or their internal physical states. The IP addresses or the MAC addresses are the unique identifiers for the things in IoT. The software components were used by these 'Things' for accessing, processing and storing sensor information, or controlling actuators connected. The IoT systems can have heterogeneous things such as fire alarms,

door alarm, lighting control devices etc. Whereas M2M system is the contrast of IoT, these types of machines having homogeneous machines within an M2M area network.

(iii) Hardware vs Software Emphasis – The M2M emphasizes more on hardware with embedded modules, the emphasis of IoT is more on software. The special software is run by the IoT devices for the collection of sensor data, analysis of data and the interfacing with the cloud through IP-based communication. The fig. 1.7 shows the various components of IoT systems including the things, the Internet, communication infrastructure and the applications.
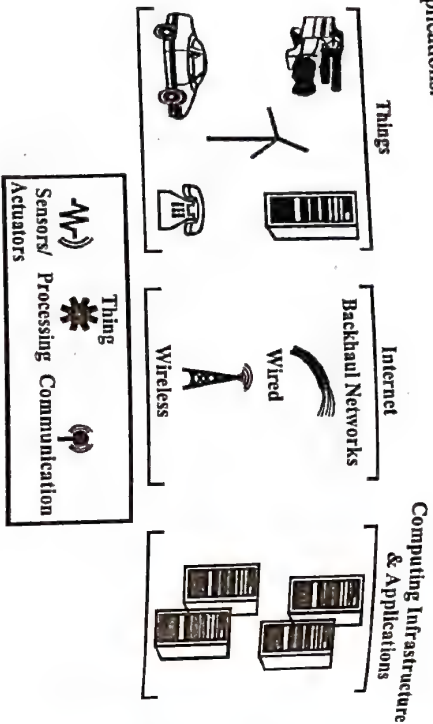

Fig. 1.7 IoT Components

(iv) Data Collection and Analysis – The collection of data in M2M is done through point solutions and often in on-premises storage infrastructure. Whereas the data in IoT is collected in the cloud, it can be public, private or hybrid cloud. The analytic component analyzes the data and stores the results in the cloud database. The data and the analysed result of the IoT are visualized with the cloud-based applications. The centralized controller is aware of the status of all the end nodes and sends control commands to the nodes. Fig. 1.8 shows the various IoT-levels and IoT components deployed in the cloud.

(v) Applications – The data is collected in M2M through point solutions and it can be accessed by on-premises applications such as service management applications, diagnosis applications and on-premises enterprise applications. The IoT data can be accessed by cloud applications such as analytics applications, enterprise applications, remote diagnosis and management applications etc.
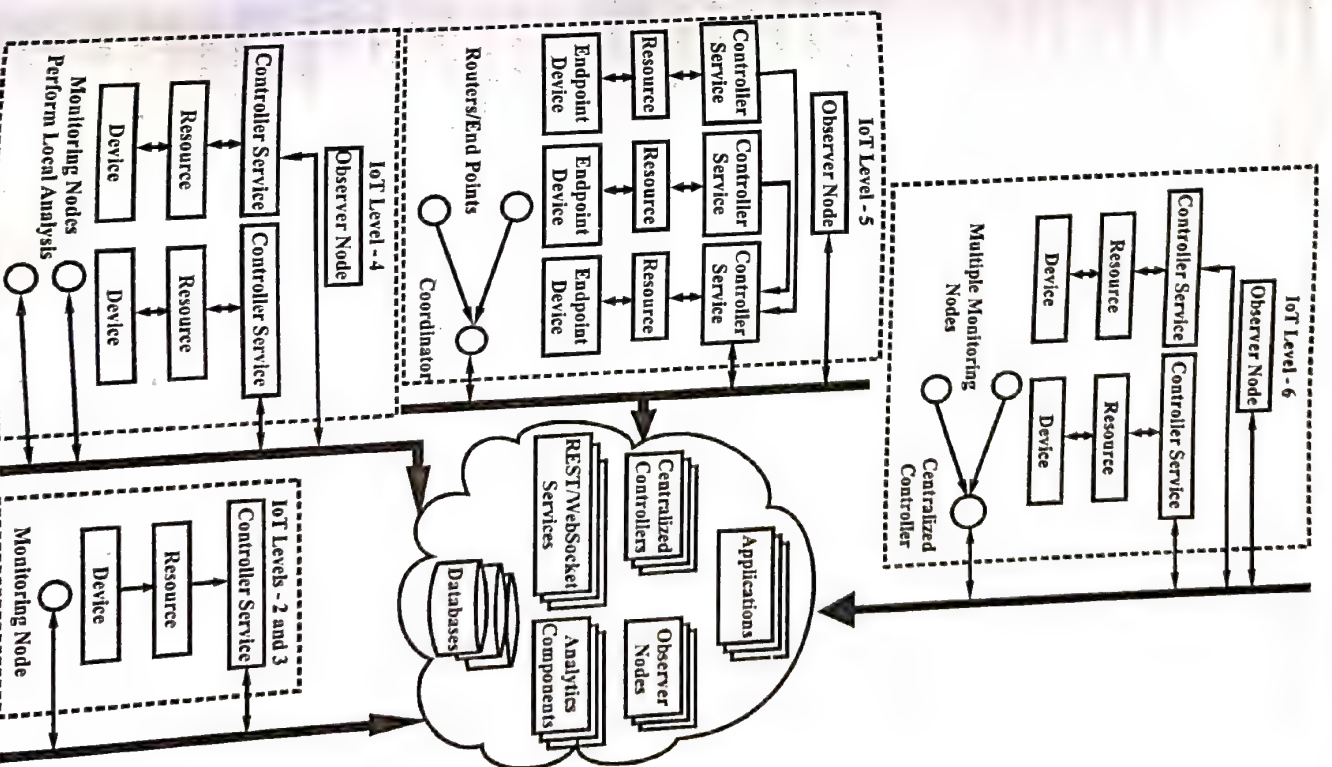

Fig. 1.8 IoT Levels and IoT Cloud Components

Q.21. *What are the differences between machines in M2M and things in IoT.*

Ans. Refer the ans. of Q.20 (ii).

Q.22. *What do you mean by Web of Things (WoT)?*

(R.G.P.V., May 2018)

Ans. Making objects which are a part of the World Wide Web is known as Web of Things (WoT). This has been made feasible by WoT software, architecture styles like JSON, REST, and programming patterns like web sockets. Data store of WoT objects is same as web pages store. The web, i.e. application layer of WoTs receives and sends data using the Internet. As RFID can uniquely identify each object on the web, the WoTs can be used for various IoT applications for the RFIDs.

Q.23. *How WoT is related with IoT?*

Ans. The Internet of Things (IoT) is a term to describe how physical objects are being connected to the Internet so that they can be explored, monitored, controlled or interacted with. The IoT is more frequently used in the context of radio frequency identification (RFID) and how physical objects are tied to the Internet and can communicate with each other. IoT systems have applications across industries via their distinctive flexibility and ability to be appropriate in any environment. They ameliorate data collection, operations, automation and much more via smart devices and powerful enabling technology. As more "things" on planet Earth are converted to the inventory of digitally connected Internet devices, the roles and accountability of web developers and technology managers will need to develop in keeping pace with the ever expanding list of appliances and gadgets that need a web interface. This global tendency is known as "Internet of Things", and as a vision has inspired that same premise for "Web of Things" and incorporates similar features and application models.

The Web of Things is a computing concept that describes a future where day-to-day objects are fully integrated with the Web. The WoT is very homogeneous to the IoT in some ways and in others it is drastically different. The stipulation for WoT is for the "things" to have embedded computer systems that enable communication with the Web. This type of smart devices would then be able to communicate with each other using current Web standards. For instance, renowned Web languages PHP, HTML, Python, and JavaScript can be used to easily build applications involving smart things and users can leverage well-known Web mechanisms such as caching, browsing, searching, and bookmarking to communicate and share these devices.

In the first instance, it endues mechanisms to formally describe IoT interfaces to permit IoT devices and services to communicate with each other,

independent of their underlying implementation, and across multiple networking protocols. In WoT, applications communicate with physical objects with the familiar HTTP protocol and RESTful API. This oversimplifies the access to physical objects, assent them to be used in web applications and merged with current web resources.

```
Applications
- OSI Layer 7
- HTTP, REST.
- CoAP, JSON
- Social Networks,
  Semantics,
  Mashup etc.
```

```
Networking
- OSI Layer 4
- EPCIS
- IPv Lowpan, Wi-Fi
  Bluetooth, ZigBee/
  IEEE802.15.4
```

Web of Things → "Application Architecture for Physical Object"

How Do We Connect Application & Things?

How Do We Connect Thing to the Internet?

Internet of Things → "Uniquely Identifiable Objects and Their Virtual Representation in an Internet Like Structure"

Fig. 1.9

Q.24. *Differentiate between IoT and WoT.*

Ans. Differences between IoT and WoT are given below –

S.No.	IoT	WoT
(i)	IoT is a network of things, which are anything that can be connected in some form to the Internet.	WoT is web network created for proper handling and using the potential of IoT platforms to provide better future.
(ii)	IoT is a hardware layer to connect everything to the Internet.	WoT is software layer to connect everything to the web.
(iii)	IoT deals with sensors, actuators, computation and communication interfaces. From a box of oranges with an RFID tag, to a smart city and to everything in between, all these digitally augmented objects make up the IoT.	WoT deals with protocols and web servers. All those applications for IoT devices make up the WoT.

(iv)	There is a different protocol for single protocol for each and every IoT devices.	WoT makes it easy by using single protocol for multiple IoT devices.
(v)	IoT platforms are hard to program due to multiple protocols.	Due to common API's to handle the protocol WoT programming is easier.
(vi)	IoT standards and prototypes are not public. They are privately founded and are not publicly accessible anytime.	WoT is free for everyone and can be accessed anywhere.
(vii)	IoT is tightly coupled between the applications and networks.	Whereas, WoT is used loosely coupled in application layer.

Q.25. Discuss about the Web of Things architecture.

Ans. Now-a-days web technology is ubiquitous and has become part of our day-to-day lives and, moreover, many tech-savvy human beings without explicit training are able to create their own web applications. Consequently, the Web of Things initiative goals for applying the well-known and proven patterns from the web to the demanding IoT domain.
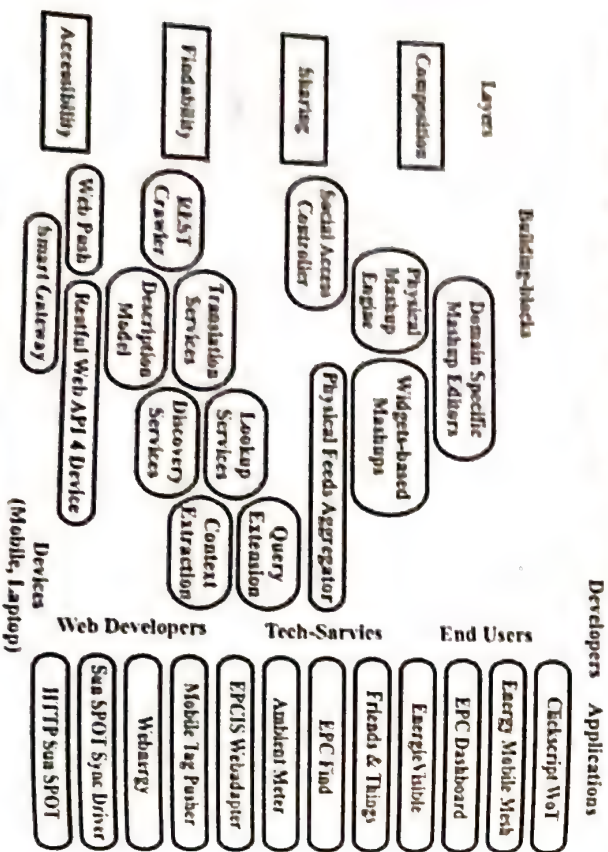

Fig. 1.10 The Web of Things Architecture

As an outcome, devices can be browsed and bookmarked. Web pages can directly include real-time data from sensors and users can build physical mashups that augment and control their day-to-day objects. There are four

layers of web of things architecture, viz., the device accessibility layer, findability layer, sharing layer and composition layer as shown in fig. 1.10. The WoT architecture is an endeavor to structure the galaxy of web protocols and tools into a useful framework for connecting any device or object to the web. Users should be able to access and use smart things absentia the need for installing additional software. From a web browser they should further have means to directly extract, share, and save smart things data and services.

This empowers creating applications in which real-world data are directly consumed by resource coercible devices, namely mobile phones or wireless sensor nodes absentia need dedicated software on these devices.

(i) Accessibility Layer – This layer is accountable for turning any thing into a web of thing that can be interacted with using HTTP requests just like any other resource on the Web. In other words, a web of thing is a REST API that permits to communicate with something in the real world. The accessibility layer in the WoT is built around two main patterns firstly all things should be exposing their services via a RESTful API, either directly or via a gateway. REST is an architectural style at the root of the programmable web thanks to its implementation in HTTP 1.1. As an outcome, if things offer RESTful APIs over HTTP, they get a URL and become seamlessly integrated with the World Wide Web and its tools like as browsers, hyperlinked HTML pages and JavaScript applications. In this context, various designs describing how the services offered by things can be accessed via REST have been proposed. Secondly, the entreaty response nature of HTTP is often cited as one of the limitations for IoT use-cases as it does not match the event-driven nature of applications that are common in the wireless sensor networks. To avoid this shortcoming while keeping a focus on fostering integration with the web, we can use HTML5 web sockets either natively or via the use of translation brokers for instance translating from MQTT or CoAP to web sockets.

(ii) Findability Layer – The web faced homogeneous challenges, when it moved from a hypertext of several thousands of documents to an application platform interconnecting an unprecedented number of documents, multimedia content and services. The focus of this layer is to provide a way to explore and locate things on the web and hence is strongly impression of the semantic web. This layer makes sure that your thing can not only be easily used by other HTTP clients, but can also be findable and automatically usable by other WoT applications. The approach here is to reuse web semantic standards to describe things and their services. This enables finding for things via search engines and other web indexes as well as the automatic generation of user interfaces or tools to communicate with things. This enables finding for things via search engines and other web indexes as well as enabling machine to machine communication based on a small set of well-defined standards and formats.

(iii) Sharing Layer – This layer specifies how the data generated by things can be shared in an efficient and secure manner over the web. At this level, another batch of web protocols assists. Firstly, TLS, the protocol that makes transactions on the web securely. Then, techniques like as delegated web authentication mechanisms like an oath which can be integrated with our things' APIs. The most basic needs for a WoT sharing platform is to be secure in order to make sure that access to smart things is not allowed to attackers. A WoT sharing platform should be uncomplicated and easy to use. Sharing the platform should also react mental models users are already familiar with. In special, it should as much as possible react the existing faith and social models of users. A WoT sharing platform should also support advertising the shared things directly on the web. In order to decrease the load of users and ameliorate security, sharing a smart thing and advertising the fact that it was shared should occur on the same channel without explicitly make known credentials.

(iv) Composition Layer – At large, the Web of Things materializes into an open ecosystem of digitally enlarge objects on top of which applications can be created using standard web languages and tools. The first three layers allow developers to access and search for web enabled smart things and owners to have a straightforward and scalable mechanism to share them. In this endmost layer, we would like to push further the boundaries of the WoT so that from getting close to developers, it also gets closer to end-users and enables them to create straightforward composite applications on top of smart things. The role of this layer is to integrate the services and data offered by things into higher level web, making it even simpler to create applications include things and virtual web services.

Q.26. What are the benefits of Web of Things ? Explain.

Ans. The benefits of Web of Things are as follows –

(i) Convenient to Program – The web protocols can easily be used to read and write data from to the devices, and are especially much simpler to use and faster to learn than the complex IoT protocols. In addition, if all devices could offer a Web API, developers could use the same programming model to interact with any of them. With the basic skills needed to build simple web applications, we can easily connect to new devices with minimal effort.

(ii) Supporting Scalability – The WoTSE must fit the search activity in local scale naturally, and at the same time, they must also be able to scale up to reach billion devices in the world. Scaling up a centralized search engine is not a preferable solution, because these systems are too far from the physical world, making them insensitive to changes. WoT makes them naturally fit for local search activity, while linking them together provides the coverage to address the upper ends of WoT scale.

(iii) Open and Extensible Standards – The reason web standards have reached such popularity is that they're entirely open and free, so there's virtually zero risk that they would change overnight. They ensure that data can be rapidly and easily moved across systems, hence HTTP and REST is an obvious choice when one wants to offer public access to some data.

(iv) Deployment, Maintenance and Integration is Rapid and Effortless – There's no risk that the web will suddenly stop working and require an upgrade. Yet, the limits of what can be done on the web have not ceased to be redefined in a decade, such as the ability to capture images from a camera or share one's location. In contrast, there are always new devices and protocols in the IoT world, and each time one of the many protocols change, all the other pieces of the puzzle that use the device need to be updated.

(v) Validating the Discovered Content – As real-world information in WoT is provided by exposing electronic tags and sensors that can be breached and forged, a malicious party can inject false information into WoT, which would be distributed by WoT search engines. A potential solution for this issue is validating the information received from sensors against past patterns and readings of their neighboring sensors. Another potential solution is building the audit-ability into WoTSE. Ensuring that one would be held accountable for his malicious activities is a powerful preventive mechanism.

(vi) Slack Coupling between Elements – The HTTP is loosely coupled by design because the contract (API specification) between actors on the web is both simple and well defined, which leaves little room for ambiguity. This allows any actor to change and evolve independently from each other (as long as the contract doesnot change). That's why you can still visit a web page that hasn't been updated since the early 90s (we'll skip any comments about its visual design). The ability for devices on the Internet of Things to talk to new devices as they get added without requiring any firmware updates is essential for a global Web of Things.

(vii) Storing and Disinfecting the Collected Data – A WoT search engine must find a balance between the number of old readings stored for resolving historical queries and building prediction models, and the scale, the resources that it manages, because each set of past measurements duplicates the whole resource collection. As a result, mechanisms for ensuring the scalability of the data storage such as distribution, deployment and purging strategies must be investigated.

Q.27. Discuss about reference IoT layered architecture.

Or

Describe IoT architectural view in detail.   (R.G.P.V., May 2018)

Ans. The reference IoT layered architecture (RILA) is shown in fig. 1.11. RILA architecture, has six layers and two cross-section layers which affect all other layers. Security and management are two cross-section layers which affect all other layers.

Security

Device Integration
(Includes different devices, measurements and actions)

Fig. 1.11 RILA Architecture

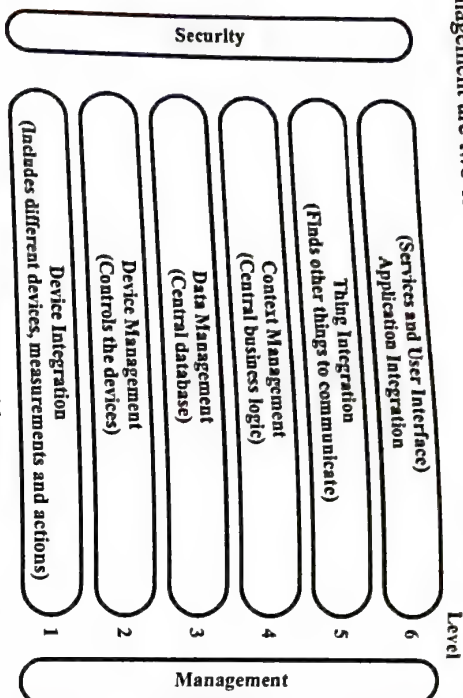| Level	
6	(Services and User Interface) Application Integration
5	Thing Integration (Finds other things to communicate)
4	Context Management (Central business logic)
3	Data Management (Central database)
2	Device Management (Controls the devices)
1	Device Integration (Includes different devices, measurements and actions)

Management

(i) ***Device Integration Layer*** – It contains various types of devices, receives their measurements and communicates actions. This layer can be appeared as a translator, that speaks many other languages like Karzel et al., 2016. The output of the sensors and tags, as well as the input of the actuators, depends on the protocol they implement.

(ii) ***Device Management Layer*** – It is responsible for receiving device registrations and sensor measurements, from the device integration layer and for communicating status changes for actuators to the device integration layer. Then, the device integration layer checks when the status change conforms with the respective actuator and translates the status change to the actuator. The device management layer controls the devices that are connected to the system. Every change to a device's registration, as well as new measurement data, should be communicated from the device integration layer to the device management layer, hence information can be updated and stored.

(iii) ***Data Management Layer*** – Data management layer stores all data of a things. It is a central database. Hence, the implementation of the data management layer strongly depends on the use case.

(iv) ***Context Management Layer*** – Context management layer explains the central business logic and it is responsible for tasks such as defining the goals of the thing, consuming and producing the context situations of the things, evaluating the context situation toward the goals, triggering actions

that will help to fulfill the goal according to the evaluated rules and in last publishing context situations for other things.

(v) ***Thing Integration Layer*** – Thing integration layer is responsible for searching other things to communicate, verifies if communication with the new thing is possible and is responsible for a registration mechanism.

(vi) ***Application Integration Layer*** – Application integration layer connects the user to the thing. It is also considered the service layer or even a simple user interface. The concrete implementation of the layer depends on the use case.

Q.28. What do you mean by IoT reference architecture ? Give its uses.

Ans. Refer to Q.27.

Beneficial uses of the IoT reference architecture are –

(i) It provides a language for everyone involved.

(ii) It provides an abstract but also rich view of the domain.

(iii) Can assist IoT project leaders in planning the work at hand and the teams needed.

IOT NETWORK CONFIGURATIONS, IoT LAN, IoT WAN, IoT NODE, IoT GATEWAY, IoT PROXY, REVIEW OF BASIC MICROCONTROLLERS AND INTERFACING

Q.29. Discuss about the IoT network configuration.

Ans. In terms of the connectivity, 'I' means the internet of computer analogously i.e. IoT LAN, IoT WAN, IoT Node, IoT Gateway and IoT Proxy connectivity terminology as shown in fig. 1.12.

(i) ***IoT LAN*** – It is very similar to the traditional LAN. It is for short range communication may be building wide or campus wide.

(ii) ***IoT WAN*** – It is basically internet-working of two different LANs and also connecting various network segments.

(iii) ***IoT Node*** – The connectivity of different nodes inside a LAN called IoT node. Sometimes this node directly connected to the Internet through the WAN.

(iv) ***IoT Gateway*** – This is a connectivity of several LANs connected together through the WAN using the gateways.

(v) ***IoT Proxy*** – It performs active application layer functions between IoT nodes and other entities.

IoT LAN
IoT WAN
IoT Node
IoT Gateway
IoT Proxy

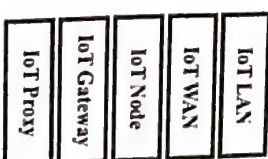**Fig. 1.12 Connectivity Terminology**

Fig. 1.13 shows various network configurations of IoT. In fig. 1.13 (a) IoT LAN has its own IoT devices and these devices has its own local address. It might happen that a particular address might be unique to this LAN, but may be reused in another LAN.
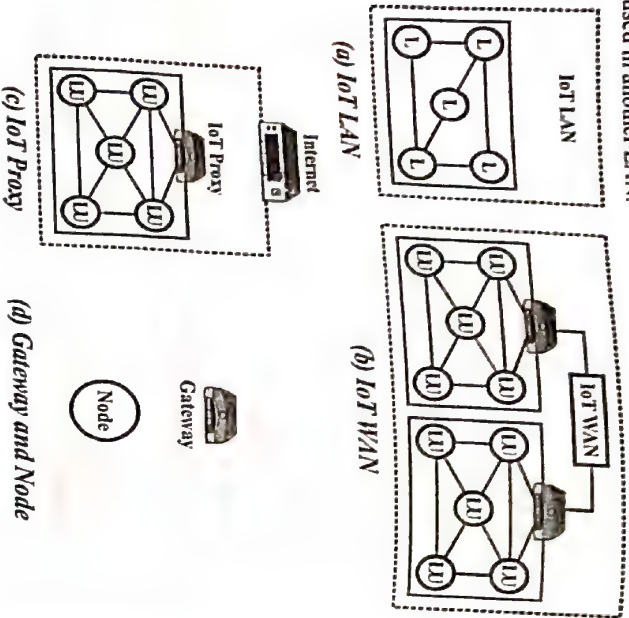

(a) IoT LAN

(b) IoT WAN

(c) IoT Proxy   (d) Gateway and Node

Fig. 1.13 IoT Network Configurations

In fig. 1.13 (b) IoT WAN is interconnection of two different LANs. These two different LANs are connected via two differents gateways. In fig. 1.13 (c) IoT proxy is basically connect to the external Internet and performs active application layer functions between node and other entities. In fig. 1.13 (d) the GATEWAY has unique network PREFIX, which can be used to identify them globally, so this strategy basically saves a lot of unnecessarily address wastage and although the NODE have to communicate to the internet via the gateway.

Q.30. Briefly describe the major network classifications.

Ans. Networks are generally classified by size, which includes geographic area, distance between stations, number of computers, transmission speed (bps), transmission media, and the network's physical architecture. The four primary classifications of networks are local area networks (LANs), metropolitan area newtorks (MANs), wide area networks (WANs), and global area networks (GANs). In addition, there are three primary types of interconnecting networks – building backbone, campus backbone, and enterprise network. Two ~~ ··· ·· ng computer networks ~f the future share

the same acronym – the PAN (personal area network) and PAN (power line area network sometimes called PLAN). The idea behind a personal area network is to allow people to transfer data through the human body simply by touching each other. Power line area networks use existing ac distribution networks to carry data wherever power lines go, which is virtually everywhere.

When two or more networks are connected together, they constitute an internetwork or internet. Fig. 1.14 illustrates the geographic relationship among computers and the different types of networks.

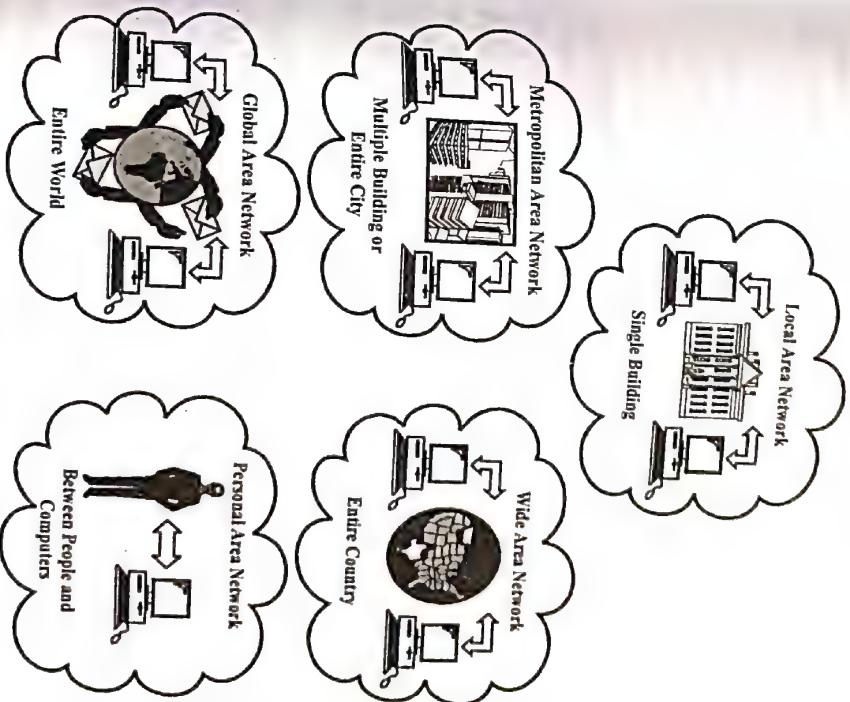**Fig. 1.14 Computer Network Types**

(i)   Local Area Network (LAN) – A local area network (LAN) is usually privately owned and links the devices in a single office, building, or campus. Depending on the requirements of an organization and the type of technology used, a LAN can be as simple as two PCs and a printer in someone's home office, or it can extend throughout a company and includes voice, sound,

and video peripherals. LANs are differentiated from other types of networks by three characteristics –

(a) Their size (b) Their transmission technology (c) Their topology.

Size of the LANs is restricted. It means that the worst-case transmission time is bounded and known in advance. By knowing this bound, it becomes possible to use certain types of designs which would not otherwise be possible. It also simplifies network management.

LANs often use a transmission technology which consists of a single cable to which all the machines are attached, like the telephone company party lines once used in rural areas. Traditional LANs run at speeds of 10 to 100 Mbps, have low delay (tens of microseconds), and make very few errors.
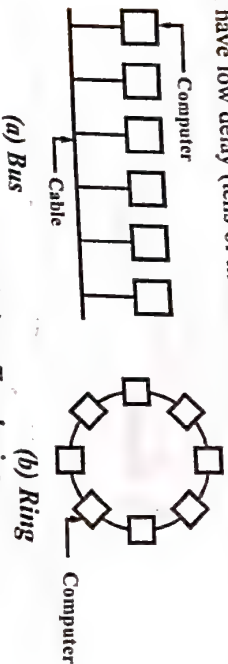

(a) Bus

(b) Ring

Fig. 1.15 Various Topologies

To broadcast LANs many topologies are possible. Two of them are shown in fig. 1.15. In a bus topology, at any moment one machine is the master and is permitted to transmit. All other machines are needed to refrain from sending. An arbitration mechanism is required to resolve conflicts when two or more machines want to transmit simultaneously. Arbitration mechanism can be centralized or distributed.

Ring is the second type of broadcast system. In a ring, each bit propagates around on its own, not waiting for the rest of the packet to which it belongs. Each bit circumnavigates the whole ring in the time it takes to transmit a few bits, often before the complete packet has even been transmitted.

Other type of LAN is built using point-to-point lines. Individual lines connect a specific machine with another specific machine. This type of LAN is really a miniature wide area network.

A local area network is generally a private owned communications network within a confined geographical area. The range is usually within a few kilometres. (less than 10 km) – may be one office, one building or a group of buildings located closely such as a college campus. The transmission channels generally use coaxial or fibre optic cables and special interface units rather than telephone lines and modems.

(ii) Metropolitan Area Network (MAN) – A metropolitan area network (MAN) is basically a bigger version of a LAN and normally uses

similar technology. It is designed to extend over an entire city. It may be a single network such as a cable television network or it may be a means of connecting a number of LANs into a larger network so that resources may be shared LAN-to-LAN as well as device-to-device. A key aspect of a MAN is that there is a broadcast medium to which all the computers are attached, due to which the design is greatly simplified as compared to other kind of networks.

A MAN can support both data and voice, and might even be related to the local cable television distribution network. It might cover a group of nearby offices or a city and might be either private or public. A MAN just has one or two cables and does not contain switching elements.

(iii) Wide Area Networks – Wide area network (WAN) provides long-distance transmission of data, voice, image and video information over large geographical areas that may comprise a country, a continent, or even the whole world. It has a collection of machines intended for running user programs. These machines are called *hosts*. Hosts are connected by a *communication subnet* or just *subnet*. The subnet carries messages from host-to-host. By separating the pure communication aspects of the network (subnet) from the application aspects (hosts), the complete network design is greatly simplified.

There are two distinct components of the subnet namely, most wide area networks which are transmission lines and switching elements. Transmission lines move bits between machines. Switching elements are specialized computers which are used to connect two or more transmission lines.

In most WANs, the network has numerous cables or telephone lines, each one connecting a pair of routers. When two routers which do not share a cable wish to communicate, they must perform this indirectly, via other routers. When a packet is transmitted from one router to another via one or more intermediate routers, the packet is received at each intermediate router in its entirety, stored there until the needed output line is free, and then forwarded. A subnet using this principle is known as a *point-to-point, store-and-forward, or packet-switched* subnet.
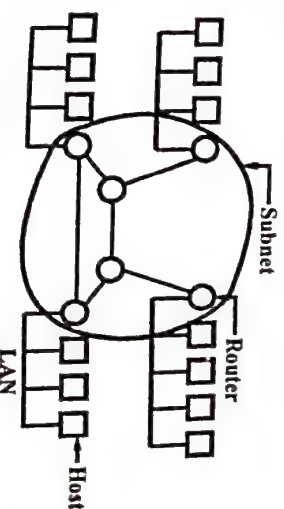

Fig. 1.16 Relation between Hosts and Subnet

A WAN can be a public system such as the Public Switched Telephone Network (PSTN) or one of the various packet switched services provided by the public telecommunication authorities. WAN can also use most other types of circuits including satellite networks, ISDN, ATM, Value Added Networks. The main distinguishing feature between a LAN and a WAN is that the LAN is under the complete control of the owner.

Various possible topologies for a point-to-point subnet are shown in fig. 1.17. Wide area networks have irregular topologies.

(a) Star   (b) Ring   (c) Tree

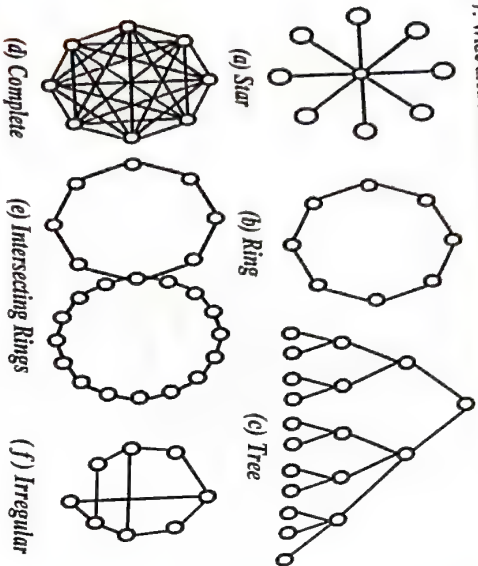(d) Complete   (e) Intersecting Rings   (f) Irregular

Fig. 1.17 Various Topologies of a Point-to-Point Subnet

The other possibility for a WAN is a satellite or ground radio station. Each router has an antenna through which it can transmit or receive. All routers can hear the output from the satellite. In some cases, they can also hear the upward transmissions of their fellow routers to the satellite as well. Sometimes routers are attached to a substantial point-to-point subnet, with only some of them having a satellite antenna.

(iv)   *Global Area Network* – Global area networks (GANs) provide connections between countries around the entire globe. The Internet is a good example of a GAN, as it is essentially a network comprised other networks that interconnect virtually every country in the world. GANs operate from 1.5 Mbps to 100 Gbps and cover thousands of miles.

(v)   *Building Backbone* – A building backbone is a network connection that normally carries traffic between departmental LANs within a single company. A building backbone generally consists of a switch or a router that can provide connectivity to other networks, such as campus backbones, enterprise backbones, MANs, WANs, or GANs.

(vi)   *Campus Backbone* – A campus backbone is a network connection used to carry traffic to and from LANs located in various buildings on campus. A campus backbone is designed for sites that have a group of buildings at a single location, such as corporate headquarters, universities, airports, and research parks.

A campus backbone normally uses optical fibre cables for the transmission of media between buildings.

(vii)   *Enterprise Networks* – An enterprise network includes some or all of the previously mentioned networks and components connected in a cohesive and manageable fashion.

Q.31. Discuss various functions supported by IoT Gateway.

Ans. The IoT gateway is believed to be an important component of the IoT network. The main functionality of the conventional gateway or router is to forward packets to the destination node. The main purpose of the IoT gateway is similar to that of a conventional gateway, but there are some additional features as follows –

First, heterogeneous network connectivity must be guaranteed. A short-range communication technology is definitely required to provide connectivity to small sensors or devices. These nodes must send packets to the IoT gateway first, so the IoT gateway must support all of these short-range communication protocols. Further, the IoT gateway must support wired communications, high-performance communication protocols including Wi-Fi and LTE, and existing short-range communication protocols including ZigBee, Bluetooth, and Z-wave. Second, network manageability is crucial for the IoT gateway. A conventional gateway manages the nodes in the subnet. However, the concept of network manageability is larger in an IoT gateway than in a conventional gateway. The IoT gateway is not limited to supporting conventional network management functions, rather, it should even update the affiliated devices' firmware or system. Consequently, the concept of manageability of the IoT gateway has expanded, and techniques to empower the network management is essential for the IoT gateway.

Finally, the IoT gateway must support the protocol or platform interworking. Many novel network concepts such as platforms and protocols, are developed to make IoT environment settle into traditional networks. Nonetheless, these can cause compatibility problems with conventional devices because legacy devices, such as household appliances, are not equipped with platforms and protocols. Thus, compatibility with the conventional platform and standard competition issues must be solved by allowing the IoT gateway to support platform interworking.

Q.32. Discuss functions of microcontrollers in IoT systems. Give examples.

Ans. IoT technologies with microcontrollers had been applied in various fields, such as wearable systems to remotely monitor the activities of users. In these systems, the main functions of the microcontrollers include data collection, data transfer, and data analysis. Moreover, the microcontrollers are also used to interconnect devices from different vendors. The frequently used microcontrollers include Arduino, Raspberry Pi microcontroller board, and an 8-bit microcontroller to power lightweight sensors. In industrial applications, the microcontrollers should be well incorporated into the IoT systems. Therefore, the logical circuits would be designed according to the real requirements. Moreover, the distributed architecture might be useful for the complex IoT systems.

Two of the most popular microcontroller based systems are Arduino and Raspberry Pi.

(i) Arduino – The Arduino Uno is a microcontroller card that supports the ATmega328. All sensors are integrated into the Arduino Uno. These sensors provide information about the ambient conditions for the Arduino Uno. Arduino Uno makes the necessary decisions/actions and uses cloud computing to inform farmers about sensor readings and necessary actions. And also send them a message with the help of GSM.

(ii) Raspberry Pi – A Raspberry Pi must be initialized by getting Raspbian, which is an operating system based on Linux. The latest version is called Debian. A Raspberry Pi board is fully capable of basic programs. The system uses the Raspberry Pi 3 Model B+ for processing, which has a GPIO and a port that is used to connect other devices. The Raspberry Pi uses an SD Card for installing Linux operating systems. SD card of minimum 2 GB is required, but it is recommended to use 4 GB or more for the system.

Q.33. Explain in brief about interfacing an LED and switch with Raspberry Pi.

Ans. **For Controlling an LED with a Switch, a Code in Python Language –**

```
from time import sleep
import RPi.GPIO as GPIO
GPIO.setmode (GPIO.BCM)
#switch Pin
GPIO.setup(26, GPIO.IN)
#LED Pin
```

```
GPIO.setup(28, GPIO.OUT)
State=false
def toggleLED (pin):
    state = not state
    GPIO.output (pin, state)
while True :
    try :
        if (GPIO.input (26) == True) :
            toggleLED (pin)
            sleep (.02)
    except keyboardInterrupt :
        exit ()
```

Code in Python Language for Sending an Email on Switch Press –

```
import smtplib
from time import sleep
import Rpi.GPIO as GPIO
from sys import exit
from_email = '<my-email>'
receipients_list = ['< receipient-email >']
cc_list = []
subject = 'Hello'
message = 'Switch pressed on Raspberry Pi'
username = '<Gmail-username>'
password = '<password>'
server = 'smtp.gmail.com:587'
GPIO.setmode (GPIO.BCM)
GPIO.setup (25, GPIO.IN)
def sendemail (from_addr, to_addr_list, cc_addr_list,
        subject, message,
        login, password,
        smtpserver):
    header = 'From : %s \n' % from_addr
    header += 'To : %s \n' % ','.join (to_addr_list)
    header += 'Cc : %s \n' % ','.join (cc_addr_list)
    header += 'Subject : %s \n' % subject
    message = header + message
```

```
server = smtplib.SMTP (smtpserver)
server.startls ()
server.login (login, password)
problems = server.sendmail (from_addr, to_addr, message)
server.quit ()
while true :
```

username, password, server)

```
try :
    sleep (.01)
    except KeyboardInterrupt :
    exit ()
if (GPIO.input (25) == True) :
sendemail (from_email, recipients_list, cc_list, subject, message,
```

Connecting an LED and switch to Raspberry Pi are represented in fig. 1.18. The first program represents a Python program for controlling an LED with a switch. In this case, the LED is attached with GPIO pin 28 and switch is attached with pin 26. In the infinite while loop, checks the value of pin 26 and toggles the state of LED when the switch is pressed. This code represents how to get input from GPIO pins and process the input and take some action. In this code, the action is toggling the state of an LED. In second code action is an email alert. The second program represents a Python program for sending an email on switch press. When switch connected to Raspberry Pi is pressed, this code uses the Python SMTP library for sending an email.
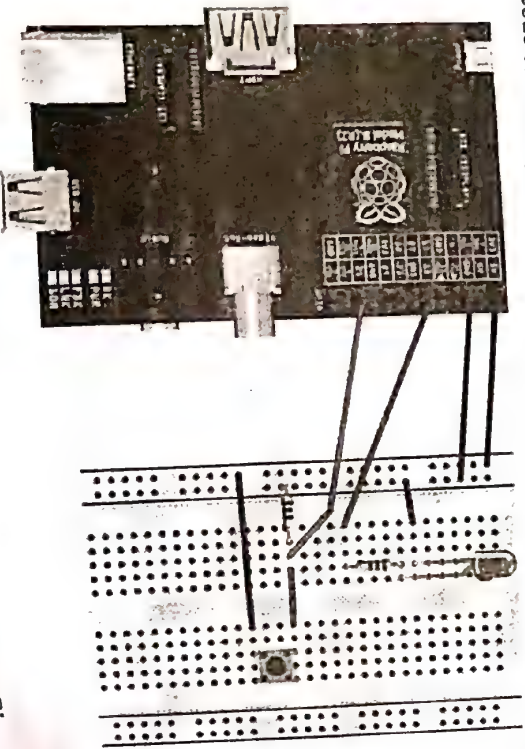

Fig. 1.18 Interfacing LED and Switch with Raspberry Pi

Q.34. Define Internet protocol (IP).

Ans. The Internet protocol (IP) is the transmission mechanism used by the TCP/IP protocols. It is an unreliable and connectionless datagram protocol – a best-effort delivery service. The term best-effort means that IP provides no error checking or tracking. IP assumes the unreliability of the underlying layes and does its best to get a transmission through to its destination, but with no guarantees.

Q.35. Write short note on IPv4.   **(R.G.P.V., June 2011)**

Ans. The Internet Protocol version 4 (IPv4) is the delivery mechanism used by the TCP/IP protocols. Fig. 1.19 shows the position of IPv4 in the suite.
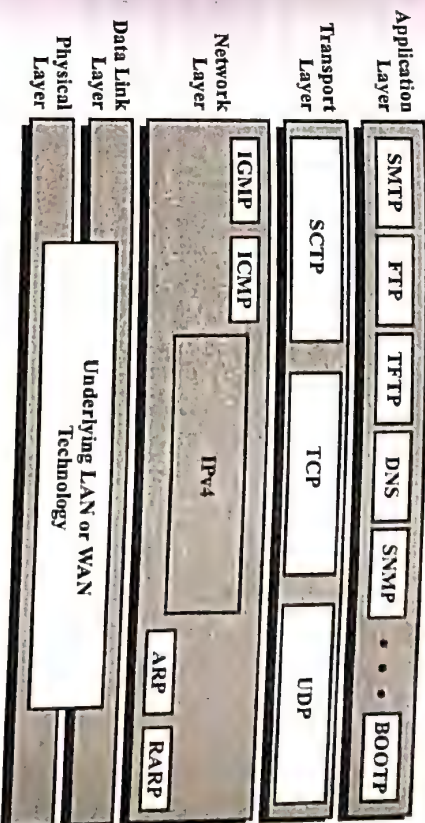
Fig. 1.19 Position of IPv4 in TCP/IP Protocol Suite

If reliability is important, IPv4 must be paired with a reliable protocol such as TCP. An example of a more commonly understood best-effort delivery service is the post office. The post office does its best to deliver the mail but does not always succeed. If an unregistered letter is lost, it is up to the sender or would-be recipient to discover the loss and rectify the problem. The post office itself does not keep track of every letter and cannot notify a sender of loss or damage.

IPv4 is also a connectionless protocol for a packet-switching network that uses the datagram approach. This means that each datagram is handled independently, and each datagram can follow a different route to the destination. This implies that datagrams sent by the same source to the same destination could arrive out of order. Also, some could be lost or corrupted during transmission. Again, IPv4 relies on a higher-level protocol to take care of all these problems.

Q.36. What are the limitations of IPv4 ?

Ans. IPv4 has following limitations –

(i)   It has limited number of IP addresses

(ii) No provision for authentication and encryption or decryption

(iii) Inefficient use of options and IP addresses

(iv) Inefficient provisions for routing

(v)   Inadequate delay and resource reservation strategies for real time audio and video transmission.

Q.37. Explain in brief about Internet protocol version 4.

Ans. The data stack received or sent to network layer and IP header fields of n (total number of header words) words are involved by IP packet. The extension will be performed when necessary actions and using data stack from or for transport layer. When a packet sents data, internet layer protocol (IP) can be returned is a process. The transmission can be defined as the unacknowledged data flow. When using IP protocol, the Internet layer receives on transfer data from transport layer to the receiver's end. PDU has maximum limit 2^{16} byte and $PDU_{IP} = 1$ packet. When using IP packet, PDU_{IP} is the maximum data unit which can sent or receive at the layer. In fig. 1.20, IP data stack comprises of IP header and at Internet layer, the data stack received or sent. IP header fields 160 bits are consisted by IP packet. When necessary, the extended header contains $(n-5)$ words, plus data stack of *len* words from or for the transport layer.
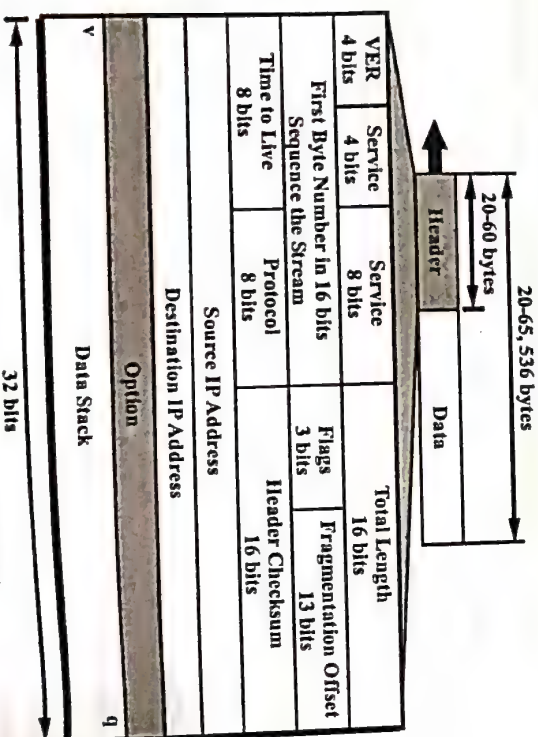

Fig. 1.20 IPv4 Datagram Format

A datagram is a variable-length packet consisting of two parts – header and data. The header is 20 to 60 bytes in length and contains information essential to routing and delivery. It is customary in TCP/IP to show the header in 4-byte sections.

Version (VER) – This 4-bit field defines the version of the IPv4 protocol. Currently the version is 4. The IPv4 software running in the processing machine that the datagram has the format of version 4. All fields must be interpreted as specified in the fourth version of the protocol.

Header Length (HLEN) – This 4-bit field defines the total length of the datagram header in 4-byte words. This field is needed because the length of the header is variable (between 20 and 60 bytes). When there are no options, the header length is 20 bytes and the value of this field is $5 (5 \times 4 = 20)$ when the option field is at its maximum size, the value of this field is $15 (15 \times 4 = 60)$.

Services – IETF has changed the interpretation and name of this 8-bit field. This field, previously called service type, is now called differentiated services we show both interpretation in fig. 1.21.

Fig. 1.21 Service Type or Differentiated Services

(i) Service Type – In this interpretation, the first 3 bits are called precedence bits. The next 4 bits are called type of service (TOS) bits, and the last bit is not used.

Precedence is a 3-bit subfield ranging from 0(000 in binary) to 7(111 in binary). The precedence defines the priority of the datagram in issues such as congestion.

TOS (Type of Service) bits is a 4-bit subfield with each bit having a special meaning. Although a bit can be either 0 or 1, one and only one of the bits can have the value of 1 in each datagram. The bit patterns and their interpretations are given in table 1.1. With only 1 bit set at a time, we can have five different types of services.

Table 1.1 Type of Service

TOS Bits	Description
0000	Normal (default)
0001	Minimize cost
0010	Maximize reliability
0100	Maximize throughput
1000	Minimize delay

(ii) *Differentiated Services* – The first 6 bits make up the codepoint
subfield and the last 2 bits are not used. The codepoint subfield can be used in
two different ways –

(a) When the 3 rightmost bits are 0s, the 3 leftmost bits are
interpreted the same as the precedence bits in the service type interpretation.
In other words, it is compatible with the old interpretation.

(b) When the 3 rightmost bits are not all 0s, the 6 bits define 64
services based on the priority assignment by the Internet or local authorities
according to table 1.2. The first category contains 32 service types; the second
and the third each contains 16 service types.

Table 1.2 Values for Codepoints

Category	Codepoint	Assigning Authority
1	XXXXX0	Internet
2	XXXX11	Local
3	XXXX01	Temporary or experimental

Q.38. Explain the functions of the three flags in the IPv4 header.

(R.G.P.V., Dec. 2012)

Ans. The three bits are known as flag bits of which two are currently
used. The first, known as the don't fragment or D bit, is intended for use by
intermediate gateways. A set D bit indicates that a network should be chosen
that can handle the datagram as a single entity rather than as multiple smaller
datagrams – known as fragments. Hence if the destination host is connected
to that network (or subnet) it will receive the user data in a single datagram or
not at all. The transmit delay of the user data can therefore be more accurately
quantified. This bit may be useful if it is known that the destination does not
have the capability to reassemble fragments. However, if this bit is set, the
datagram will be discarded if it exceeds the maximum size of an enroute
network. Therefore, if the bit is set, it may be advisable to use source routing
to avoid networks with small maximum packet size.

The second flag bit, known as more fragments or M bit, is used during
the assembly procedure associated with user data transfers involving multiple
datagrams.

Q.39. How is the IPv4 header checksum calculated ? (R.G.P.V., Dec. 2012)

Ans. Checksum is the primary means of errors detection used by protocols
in the TCP/IP protocol suite. The checksum provides error checking on the IP
header only. The IP header checksum is calculated on the number of bytes
specified in the header length field. However, the receiver performs the
checksum calculation on the entire header, including the checksum.

The calculation of checksum at the sending side is done as follows –

(i) The checksum is made all 0s.

(ii) The datagram is divided into k groups of n bits each, where n is
usually 16.

(iii) The 16-bit groups are added together using one's complement
arithmetic, which produces an n-bit sum.

(iv) The sum is complemented and placed in the checksum field.

The calculation of checksum at the receiving side is done as follows –

(i) The received datagram is divided into k groups of n bits each.

(ii) The groups are added together.

(iii) The sum is complemented.

(iv) If the result is all 0s, the packet is assumed to be error free;
otherwise, the packet is rejected.

Q.40. Define IPv6.

Ans. IPv6, also known as IPng (IP next generation), uses 128-bit (16
byte) addresses, versus the 32-bit (4-byte) addresses currently used in version
4. IPv6 can nearby accommodate a larger number of user. In version 6, the
packet format has been simplified, yet at the same time it is more flexible to
allow for the future addition of features. The new version support authentication,
data integrity, and confidentiality at the network layer. It is designed to handle
the transmission of real-time data such as audio and video, and can carry data
from other protocols. IPng can also handle congestion and route discovery
better than the current version.

Q.41. List out various advance features of IPv6 as compared to IPv4.

(R.G.P.V., June 2010)

Ans. The major features (changes) introduced by IPv6 can be grouped
into seven categories –

(i) *Larger Addresses* – The new address size is the most noticeable
change. IPv6 quadruples the size of an IPv4 from 32 to 128 bits.

(ii) *Extended Address Hierarchy* – IPv6 uses the larger address
space to create additional levels of addressing hierarchy.

(iii) *Flexible Header Format* – IPv6 uses an entirely new and
incompatible datagram format. Unlike the IPv4 fixed-format header, IPv6
defines a set of optional headers.

(iv) *Improved Options* – Like IPv4, IPv6 allows a datagram to include
optional control information. IPv6 includes new options that provide additional
facilities not available in IPv4.

(ii) Differentiated Services – The first 6 bits make up the codepoint subfield and the last 2 bits are not used. The codepoint subfield can be used in two different ways –

(a) When the 3 rightmost bits are 0s, the 3 leftmost bits are interpreted the same as the precedence bits in the service type interpretation. In other words, it is compatible with the old interpretation.

(b) When the 3 rightmost bits are not all 0s, the 6 bits define 64 services based on the priority assignment by the Internet or local authorities according to table 1.2. The first category contains 32 service types; the second and the third each contains 16 service types.

Table 1.2 Values for Codepoints

Category	Codepoint	Assigning Authority
1	XXXXX0	Internet
2	XXXX11	Local
3	XXXX01	Temporary or experimental

Q.38. Explain the functions of the three flags in the IPv4 header.

(R.G.P.V., Dec. 2012)

Ans. The three bits are known as flag bits of which two are currently used. The first, known as the don't fragment or D bit, is intended for use by intermediate gateways. A set D bit indicates that a network should be chosen that can handle the datagram as a single entity rather than as multiple smaller datagrams – known as fragments. Hence if the destination host is connected to that network (or subnet) it will receive the user data in a single datagram or not at all. The transmit delay of the user data can therefore be more accurately quantified. This bit may be useful if it is known that the destination does not have the capability to reassemble fragments. However, if this bit is set, the datagram will be discarded if it exceeds the maximum size of an enroute network. Therefore, if the bit is set, it may be advisable to use source routing to avoid networks with small maximum packet size.

The second flag bit, known as more fragments or M bit, is used during the assembly procedure associated with user data transfers involving multiple datagrams.

Q.39. How is the IPv4 header checksum calculated? (R.G.P.V., Dec. 2012)

Ans. Checksum is the primary means of errors detection used by protocols in the TCP/IP protocol suite. The checksum provides error checking on the IP header only. The IP header checksum is calculated on the number of bytes specified in the header length field. However, the receiver performs the checksum calculation on the entire header, including the checksum.

The calculation of checksum at the sending side is done as follows –

(i) The checksum is made all 0s.

(ii) The datagram is divided into k groups of n bits each, where n is usually 16.

(iii) The 16-bit groups are added together using one's complement arithmetic, which produces an n-bit sum.

(iv) The sum is complemented and placed in the checksum field.

The calculation of checksum at the receiving side is done as follows –

(i) The received datagram is divided into k groups of n bits each.

(ii) The groups are added together.

(iii) The sum is complemented.

(iv) If the result is all 0s, the packet is assumed to be error free; otherwise, the packet is rejected.

Q.40. Define IPv6.

Ans. IPv6, also known as IPng (IP next generation), uses 128-bit (16 byte) addresses, versus the 32-bit (4-byte) addresses currently used in version 4. IPv6 can nearby accommodate a larger number of user. In version 6, the packet format has been simplified, yet at the same time it is more flexible to allow for the future addition of features. The new version support authentication, data integrity, and confidentiality at the network layer. It is designed to handle the transmission of real-time data such as audio and video, and can carry data from other protocols. IPng can also handle congestion and route discovery better than the current version.

Q.41. List out various advance features of IPv6 as compared to IPv4.

(R.G.P.V., June 2010)

Ans. The major features (changes) introduced by IPv6 can be grouped into seven categories –

(i) **Larger Addresses** – The new address size is the most noticeable change. IPv6 quadruples the size of an IPv4 from 32 to 128 bits.

(ii) **Extended Address Hierarchy** – IPv6 uses the larger address space to create additional levels of addressing hierarchy.

(iii) **Flexible Header Format** – IPv6 uses an entirely new and incompatible datagram format. Unlike the IPv4 fixed-format header, IPv6 defines a set of optional headers.

(iv) **Improved Options** – Like IPv4, IPv6 allows a datagram to include optional control information. IPv6 includes new options that provide additional facilities not available in IPv4.

(v) *Provision for Protocol Extension* – Perhaps the most vital change in IPv6 is a move from a protocol that fully specifies all details to a protocol that permit additional features. The extension capability has the potential to allow the IETF to adopt the protocol to changes in underlying network hardware or to new applications.

(vi) *Support for Autoconfiguration and Renumbering* – IPv6 provides facilities that allow computers on an isolated network to assign themselves addresses and begin communicating without depending on a router or manual configuration. The protocol also includes a facility that permits a manager to renumber networks dynamically.

(vii) *Support for Resource Allocation* – IPv6 has two facilities that permit preallocation of network resources, a flow abstraction and a differentiated service specification.

Q.42. Differentiate IPv4 and IPv6. (R.G.P.V., Dec. 2009)

Ans. Differences between IPv4 and IPv6 are given below –

S.No.	IPv4	IPv6
(i)	Source and destination addresses are 32 bits (4 bytes) in length.	Source and destination addresses are 128 bits (16 bytes) in length.
(ii)	IP sec support is optional.	IP sec support is required.
(iii)	IPv4 header does not identify packet flow for QoS handling by routers.	IPv6 header contains flow label field, which identifies packet flow for QoS handling by router.
(iv)	Both routers and the sending host fragment packets.	Only the sending hosts fragment packets; routers do not.
(v)	Header includes a checksum.	Header does not include a checksum.
(vi)	Header includes options.	All optional data is moved to IPv6 extension headers.
(vii)	Address resolution protocol (ARP) uses broadcast ARP request frames to resolve an IP address to a link-layer address.	Multicast neighbour solicitation message resolves IP addresses to link-layer addresses.
(viii)	Internal Group Management Protocol (IGMP) manages membership in local subnet groups.	Multicast listener discover (MLD) messages manage membership in local subnet groups.
(ix)	ICMP router discover is used to determine the IPv4 address of the best default gateway and it is optional.	ICMPv6 router soliciation and router advertisement messages are used to determine the IP address of the best default gate-way and they are required.

(x) Broadcast addresses are used to send traffic to all nodes on a subnet. | IPv6 uses a link-local scope all-nodes multicast addresses.

(xi) Must be configured either manually or through DHCP. | Does not require manual configuration or DHCP.

(xii) Uses host address (A) resource records in domain name system (DNS) to map host names to IPv4 addresses. | Uses host address (AAAA) resource records in DNS to map host names to IPv6 addresses.

(xiii) Uses pointer (PTR) resource records in the IN-ADDR. ARPA DNS domain to map IPv4 addresses to host names. | Uses pointer resource records, in the IP$_6$. ARPA DNS domain to map IPv6 addresses to host names.

(xiv) Must support a 576 byte packet size (possibly fragmented). | Must support a 1280 byte packet size (without fragmentation).

Q.43. What is the general format of an IPv6 datagram ? Discuss its header.

Ans. An IPv6 protocol data unit (known as packet) has the following general form –

Fig. 1.22

The only header that is required is referred to simply as the IPv6 header. This is of fixed size with a length of 40 octets, compared to 20 octets for the mandatory portion of the IPv4 headers.

The IPv6 header has a fixed length of 40 octets, consisting of the following fields (see fig. 1.23).
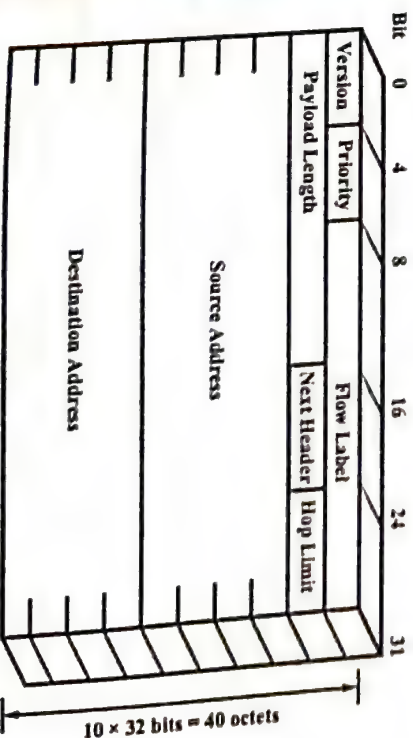

Fig. 1.23 IPv6 Header

(i) Version (4 bits) – This field tells the IP software running in the processing machine that the datagram has the format of version 6.

(ii) Priority (4 bits) – The 4-bit priority field defines the priority of the packet with respect to traffic congestion.

(iii) Flow Label (24 bits) – May be used by a host to label those packets for which it is requesting special handling by routers within a network.

(iv) Payload Length (16 bits) – Length of the remainder of the IPv6 packet following the header, in octets. This is the total length of all of the extension headers plus the transport-level PDU.

(v) Next Header (8 bits) – Identifies the type of header immediately following the IPv6 header.

(vi) Source Address (128 bits) – The address of the originator of the packet.

(vii) Hop Limit (8 bits) – The remaining number of allowable hops for this packet. The hop limit is set to some desired maximum value by the source and decremented by 1 by each node that forwards the packet.

(viii) Destination Addresses (128 bits) – The address of the intended recipient of the packet. This may not in fact be the intended ultimate destination if a routing header is present.

Q.44. Explain with example IP addressing in IoT.(R.G.P.V., May 2018)

Ans. For addresses, the designers of TCP/IP choose a scheme analogous to physical network addressing in which each host on the Internet is assigned as 32-bit integer address called its *Internet address or IP address.*

It is a unique number helps to identify each and every computer on Internet. IP addresses are in the format xxx.xxx.xxx.xxx, where each xxx is a number from 0 to 255. IP address is an identifier for a particular machine on a particular network. IP addresses are also referred to as IP numbers and Internet addresses. IP address consists of four sections separated by periods and each section has numbers from 0 to 255. These sections show machine itself or host both and the network that the host is on.

The network portion of IP address is allocated to ISPs by interNIC, under the authority of IANA (Internet Assigned Numbers Authority). Then ISPs assign the host portion of IP address to the machines on networks they operate. Which sections of IP address show the network and which sections show the machine depend on the class of IP address, that is assigned to the network.

There are five classes of IP addresses – Class A, Class B, Class C, Class D, and Class E. The classes correspond either to the size of network (i.e., number of hosts that the network can support) or are reserved of specific purposes, such as multicasting and experimentation.

The example of class A, class B and class C IP addresses are compared as –

Class A	**202.0**.0.0
Class B	**127.54**.0.0
Class C	**234.154.222**.0

Here, the bold numbers represent the network and the normal numbers represent hosts on the network. Thus, the network of Class A can support many more hosts than the network of Class C.

Class D is reserved for *multicast addresses*. Multicasting allows copies of a datagram to passed to a select group of hosts rather than to an individual host. Class E addresses are reserved for future use.

Each Internet address consists of four bytes (32 bits), defining three fields – class type, netid, and hostid. These parts are of varying lengths, depending on the class of the address (see fig. 1.24).

Class			
Type	Netid	Hostid	

Fig. 1.24 Internet Address

The different classes are designed to cover the needs of different types of organizations. For example, class A addresses are numerically the lowest. They use only one byte to identify class type and netid, and leave three bytes available for hostid numbers. This division means that class A networks can accommodate for more hosts than can class B or class C network which provide two-and one-byte hostid fields, respectively. Currently both class A and class B are full and addresses are available in class C only.

Fig. 1.25 shows the form of each IP address class.

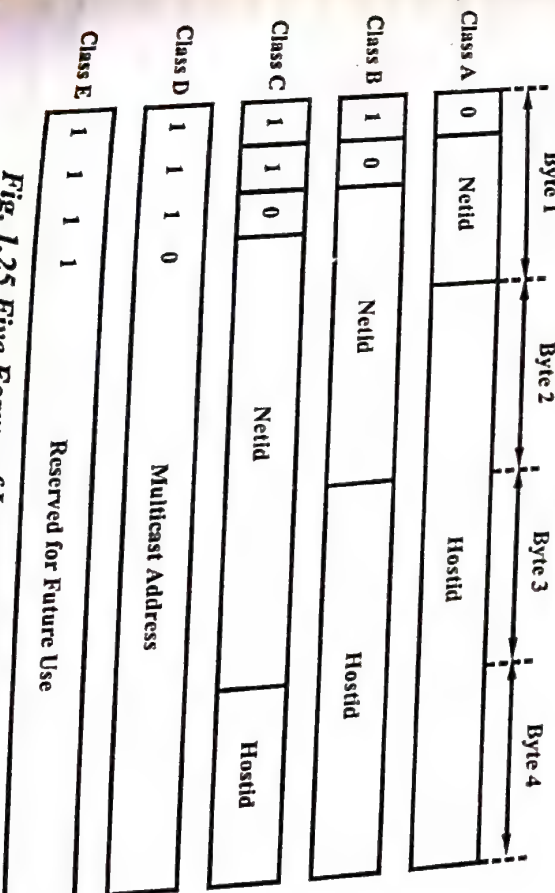|	Byte 1	Byte 2	Byte 3	Byte 4
Class A	0 Netid	Hostid	Hostid	Hostid
Class B	1 0 Netid	Netid	Hostid	Hostid
Class C	1 1 0 Netid	Netid	Netid	Hostid
Class D	1 1 1 0 Multicast Address			
Class E	1 1 1 1 Reserved for Future Use			

Fig. 1.25 Five Forms of Internet (IP) Address

Now-a-days, classless inter-domain routing (CIDR) scheme is employed. Class C 198.136.50.0/3 means class C assigned three IP addresses for Internet routing by 3 public domain servers. For the same domain name, the system administrator at a company assigns servers for each of the three IP addresses. There are two types of IP address.

    (i) Static IP address   (ii) Dynamic IP address.

    (i) Static IP Address – Static IP address is alloted by internet service provider. The service provider can give individual IP address. Service provider can give a class C network address comprising of a group of 254 IP address, when a company has a number of hosts.

    (ii) Dynamic IP Address – It requires to be assigned an individual IP address when a device connects to the Internet. When device and router are connected, the router and device both employ DHCP. At every instance, DHCP allocates an IP address to the device. This address can be referred to as dynamic IP address. When device and router are disconnected then, the dynamic IP address is lost and when device and router are reconnected then DHCP allocates new IP address to that device.

Q.45. What is IP addressing ? Why IPv6 are required to implement the concept of IoT ?

                                                     (R.G.P.V., May 2019)

    Ans. IP Addressing – Refer to Q.44.

    Version 6 (IPv6) permits for a huge increase in the number IP addresses. With IPv4, the maximum number of unique addresses, 4.2 billion, is not enough to provide even one address for each of the 7.3 billion people on Earth. IPv6, in contrast, will accommodate over 10^{38} addresses—more than a trillion per person.

    It is highly likely that to accommodate the anticipated growth in the numbers of Internet-connected objects, IPv6 will have to be implemented broadly. It has been available since 1999 but was not formally launched until 2012. In most countries, fewer than 10% of IP addresses were in IPv6 as of September 2015. Adoption is highest in some European countries and in the United States, where adoption has doubled in the past year to about 20%. Globally, adoption has doubled annually since 2011, to about 7% of addresses in mid-2015. While growth in adoption is expected to continue, it is not yet clear whether the rate of growth will be sufficient to accommodate the expected growth in the IoT. That will depend on a number of factors, including replacement of some older systems and applications that cannot handle IPv6 addresses, resolution of security issues associated with the transition, and availability of sufficient resources for deployment.

    Efforts to transition federal systems to IPv6 began more than a decade ago. According to estimates by NIST, adoption for public-facing services has

---

been much greater within the federal government than within industry or academia. However, adoption varies substantially among agencies, and some data suggest that federal adoption plateaued in 2012. Data were not available for this report on domains that are not public-facing, and it is not clear whether adoption of IPv6 by federal agencies will affect their deployment of IoT applications.

Q.46. Write short note on DNS.

                        Or

Write short note on DNS in the Internet. **(R.G.P.V., June 2010)**

                        Or

                (R.G.P.V., June 2011, 2013, 2014)

What is the utility of DNS ?

                            (R.G.P.V., June 2015)

    Ans. The essence of DNS is the invention of a hierarchical, domain-based naming scheme and a distributed database system for implementing the naming scheme. The primary use of DNS is for mapping host names and e-mail destinations to IP addresses but can also be used for other purposes.

    The DNS is a distributed database. It is used by TCP/IP application. We use the term distributed because no single site on the Internet knows all the information. Each site maintains its own database of information and runs a server program that other system across the Internet (clients) can query. The DNS provides the protocols that allow client and server to communicate with each other.

    On Unix hosts the resolver is accessed primarily through two library functions, gethostbyname(3) and gethostbyaddr(3), which are linked with the application when the application is built. The first takes a hostname and returns an IP address, and the second takes an IP and looks up a hostname. The resolver contacts one or more name servers to do the mapping.

    The resolver is normally the part of the application. It is not part of the operating system kernel as are the TCP/IP protocols.

    RFC 1034 specifies the concept and facilities provided by DNS, and RFC 1035 details the implementations and specification.

Q.47. Define the following terms with respect to IoT –

    (i) *IP addressing*   (ii) *DNS*

    (iii) *Static IP address*   (iv) *Dynamic IP address.*

                            (R.G.P.V., Nov. 2019)

    Ans. (i) *IP Addressing* – Refer to Q.44.

    (ii) *DNS* – Refer to Q.46.

    (iii) *Static IP Address* – Refer to Q.44 (i).

    (iv) *Dynamic IP Address* – Refer to Q.44 (ii).

Q.48. Discuss IPv6 addresses.

Ans. Like an IPv4 address, an IPv6 address uniquely identifies a computer on the Internet. However, the IPv6 addressing scheme differs from the IPv4 addressing scheme in many ways, as described below.

The fundamental difference between IPv6 addressing and IPv4 addressing is the address length. Whereas IPv4 works with 32-bit IP addresses, IPv6 uses 128-bit IP addresses, making the address space extremely large.

Like IPv4, IPv6 also defines an addressing format that consists of a network id and a host id. However, inlike IPv4, IPv6 does not define any classes. Thus, it is not possible to determine how many bits are reserved for the network id portion, and how many bits are reserved for the host id portion by looking at the IP address anymore. Thus, an additional *prefix length* is needed to know this fact.

Furthermore, IPv6 defines three types of addresses unicast, anycast, and multicast. A *unicast address* defines a single computer. The packet sent to a unicast address must be delivered to that specific computer.

An *anycast address* defines a group of computers with addresses that have the same prefix. For example, all computers connected to the same physical network share the same prefix address. A packet sent to one anycast address must be delivered to exactly one of the members of the group which is closest or the most easily accessible.

A *multicast address* defines a group of computers that may or may not share the same prefix and may or may not be connected to the same physical network. A packet sent to a multicast address must be delivered to each member of the set.

Since IPv6 addresses can be very long to read or write, the dotted decimal notation used in IPv4 is not used here. Instead, a new hexadecimal colon notation is used to write IPv6 addresses. In this notation, 128 bits are divided into eight sections, each 2 bytes in length. Two bytes in hexadecimal need four hexadecimal digits. Therefore, the address consist of 32 hexadecimal digits, with every four digits separated by a colon.

Example of such a notation is an address 68DC : 7764 : FFFF : FFFF : 0 : 1267 : 8C0A : FFFF.

Further optimizations of the IPv6 addressing are possible one of them is to drop leading zeroes. Suppose that a portion of an IPv6 address is 00F6 : 0089. Then, it is written as F6 : 89. Another optimization is to replace an address portion containing zero with another colon. Thus, if an address is 65CB : 0 : 1356, we can write it as 65CB : 1356.

---

also.

Q.49. What is the use of IPv6 in IoT ? Explain its power and conclusion also.

(R.G.P.V., Nov. 2019)

Ans. The uses of IPv6 in IoT are as follows –

(i) Scalability – IPv6 provides a highly scalable address scheme with 2^{128} unique addresses representing 3.4×10^{38} addresses.

(ii) Enabling the Extension of the Internet and the Web of Things – IPv6 enables the extension of the Internet to any device and service. Experiments have demonstrated the successful use of IPv6 addresses to large scale deployments of sensors in smart buildings, smart cities and even cattle.

(iii) IPv6 Version Available for Low-power Devices – The use of IPv6 for IoT applications has been investigated for many years. One of the main outcomes from research in this area is a compressed version of IPv6 for low power devices, namely IPv6 over low power wireless personal area networks (6LoWPAN). It is a simple and efficient mechanism that allows to shorten the IPv6 address size for constrained devices, while enabling border routers to translate these compressed addresses into regular IPv6 addresses.

(iv) Multicast and Anycast – With IPv6, the use of multicast is much more risk-free thanks to the way of creating the addresses for destinations groups. In fact, IPv6 defines several multicast addresses (i.e., FF01 :: 1) for the Internet auto configuration procedures.

IP multicast is particularly useful in the IoT environment. In large IoT deployments, it allows to distribute one command or set of data to all the devices directly or indirectly on the Wireless Area Network (WAN).

Anycast allows to verify if devices or other resources are available in the network. This provides a convenient mechanism for accessing both resource directory and resource depository entities. It makes the use of alternate resources much easier, thus improving the resilience of the system. Anycast is very useful in local and sensor networks. It may be used for IoT resource repositories, security servers and multi-homed gateways.

(v) Mobility – IPv6 provides strong features and solutions to support mobility of both end-nodes, and routing nodes (and thus, for resources or agents in IoT applications).

(vi) Security – To overcome security limitations of IPv4, new features have been included in IPv6. Among the features that support or improve security which can be mentioned are –

(a) Introduction of IPSec, designed for IPv6 due to restoration of end to end connectivity.

routability.

(b) Mandatory use of IPsec for Mobile IPv6 to secure the retm

(c) Large addressing space.

(d) Neighbor discovery.

(vii) Improving Routing – IPv6 provides end-to-end connectivity, with a more distributed routing mechanism. The IPv6 protocol makes routing more efficient and hierarchical by reducing the size and complexity of routing tables.

(viii) StateLess Address AutoConfiguration (SLAAC) – IPv6 provides an address self-configuration mechanism (stateless mechanism). The nodes can define their addresses in very autonomous manner. A router sends the prefix of the local link in its router advertisements. A host can generate its own IP address by appending its link-layer (MAC) address, converted into Extended Universal Identifier (EUI) 64-bit format, to the 64 bits of the local link prefix.

●●

UNIT 2

SENSOR, BASIC COMPONENTS AND CHALLENGES OF A SENSOR NODE, SENSOR FEATURES, SENSOR RESOLUTION

Q.1. What are sensors ?

Ans. A sensor is an electronic device connected in an electronic circuit, which senses a physical or real environment. A change in the specific physical condition or environment, causes a measurable change in a characteristic circuit parameter, which is sensed by a sensor. The sensor sends signals to an electronic circuit, which is connected to a microcontroller or computing device through a serial port interface.

Q.2. What is sensor technology ?

Ans. A technique used for designing sensors and associated circuits, devices and electronic readers is known as sensor technology. A change in physical parameters like pressure, light, type of metal, temperature, smoke, proximity to an object can be sensed by a sensor. Acceleration, orientation, location, vibrations, or smell, organic vapours or gases can also be sensed by sensors. Physical energy like heat, strain, sound, pressure, vibrations and motion are converted into electrical energy by a sensor. An electronic circuit connecting the input and the sensor, also receives the output of the sensor. A smart sensor besides having electronic circuit also has computing and communication ability. Variations in the parameters like currents, voltages, phase angles or frequencies provide energy to the electronic circuits. These variations are measured by analog sensors with reference to a normal or reference condition to determine the value of the sensed parameter.

Q.3. Explain the concept of sensor node architecture.

Ans. A sensor network is made up of the following parts, namely a set of sensor nodes which are distributed in a sensor field, a sink which communicates with the task manager via Internet interfacing with users. A set of sensor nodes is the basic component of a sensor network. A sensor node is composed

of four basic components as shown in fig. 2.1. They are a sensing unit, a processing unit, a communication unit and a power unit.
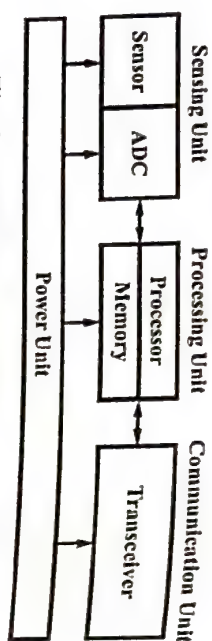

Fig. 2.1 The Components of a Sensor Node

Sensing units are usually made up of application specific sensors and ADCs (analog to digital converters), which digitalize the analog signals produced by the sensors when they sensed a particular phenomenon. In some cases, an actuator is also needed.

Obviously sensors play a key role in a sensor network which are the very front end connecting our physical world to the computational world and the Internet. Although MEMS technology has been making steady progress in the past decades, there is still large space for the further development of smart front end sensors. Among them, various chemical and biochemical sensors remain one of the most challenging sensor groups to be explored and developed, e.g. sensors to detect toxic or explosive trace in public areas, sensors for diagnostic analysis and sensors used under extreme conditions. New sensing principle, new sensing material and new sensor design need to be invented and adopted.

The processing unit is usually associated with an embedded operating system, a microcontroller and a storage part. It manages data acquisition, analyzes the raw sensing data and formulates answers to specific user requests. It also controls the communication and performs a wide variety of application specified tasks. Energy and cost are two key constraints for processing components. Nodes may have different types of processors for certain specific tasks. For example, a video sensor node may need a more powerful processor to run than a common temperature sensor. A small embedded operation system such as Berkeley's TinyOS is another key issue for an embedded system. Besides the basic ability for process management and resource management, it may also possess the capability for software tailor and real time management, the ability to provide support for embedded middleware, network protocols and embedded database.

The transceiver connects the sensor node to the network. Usually each of the sensor nodes has the capability to transmit data to and receive data from another node and the sink. The later may further communicate with the task manager via Internet (or Satellite) and information reaches the end user. A

transceiver is the most power-consuming component of the node. Thus the study of multi-hop communications and complex power saving modes of operation, e.g. having multiple different sleep states, is crucial in this content.

The power unit delivers power to all the working parts of the node. Because of the limited capacity of the power unit, e.g. the limited lifetime of a battery, the development of the power unit itself and the design of a power saving working mode of the sensor network remain some of the most important technical issues. For some applications, a solar battery may be used.

Additionally, a sensor node may have application dependent functional subunits such as a location finder, a mobilizer, a power generator and other special-purpose sensors. The nature or number of such subunits may vary, depending on the application needs.

Q.4. What are the challenges of sensor node ?

Ans. The challenges of sensor node are as follows –

(i) Sensor node relies only on battery and it cannot be recharged or replaced. Hardware design for sensor node should also be considered.

(ii) Achieving synchronization between nodes is also another issue.

(iii) Node failure, topology changes, adding of nodes and deletion of nodes is another challenging issue in sensor node.

(iv) It comes under fewer infrastructures and also maintenance is very difficult.

Q.5. What are the features of sensor ?

Ans. The features of sensor are as follows –

(i) It is only sensitive to the measured property (e.g., a temperature sensor senses the ambient temperature of a room).

(ii) Sensor is insensitive to any other property likely to be encountered in its application (e.g., a temperature sensor does not bother about light or pressure while sensing the temperature).

(iii) Sensor does not influence the measured property (e.g., measuring the temperature does not reduce or increase the temperature).

Q.6. What do you mean by sensor resolution ?

Ans. The resolution of the sensor is basically defined as the smallest change that it can detect in the quantity that is being measured. The resolution of a sensor with a digital output is usually the smallest resolution of the digital output it is capable of processing. A sensor's accuracy does not depend upon its resolution.

SENSOR CLASSES – ANALOG, DIGITAL, SCALAR, VECTOR SENSORS, SENSOR TYPES, BIAS, DRIFT, HYSTERESIS ERROR, QUANTIZATION ERROR

Q.7. Name different sensor classes.

Ans. Various sensor classes are shown in fig. 2.2. The sensor classes can be classified into two types based on output and data types. On the basis of output they can be divided as analog sensor and digital sensor. On the basis of data types they can be divided as scalar sensor and vector/ multimedia sensor.
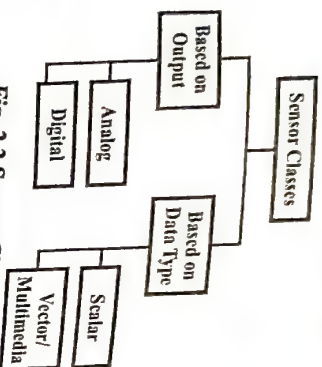

Fig. 2.2 Sensor Classes

Q.8. How an analog sensor works ?

Ans. An analog sensor consists of a sensing component and associated electronic analog circuit. Depending upon the change in the physical environmental parameters like temperature, strain, pressure, force, flex, magnetic field, or proximity, analog sensors generates the analog outputs. Resistance of sensing component of these sensors varies with the surrounding pressure or strain or magnetic field or humidity. For example, the resistance of a pressure sensor increases with the pressure which creates a strain on the sensor.

The analog output from a sensor circuit is given as input to a signal conditioning-cum-amplifying circuit (SC). The output of SC is inputted to an analog-to-digital converter (ADC). The ADC gives a digital output which can be read by a microcontroller. The microcontroller after performing computation, gives the value of the sensed parameter and shows the physical situation around the sensor.

Q.9. Write a short note on digital sensors.

Ans. Digital sensors use a special electronic circuit, which gives digital output 1 or 0 (on-off state) or output of 1s and 0s as a binary number (corresponding to a set of on-off states). In a microcontroller, the output 1 or 0 is read by a port. Digital sensors are used for sensing a sudden change in specific physical state or condition or in a specific set of physical states.

The concept of digital output of on-off state found application in multiple conditions detection, such as for sensing of presence of traffic on a street and sensing the presence of organic vapours or gas leakage or fire, etc.

Q.10. How temperature can be measured using resistance sensor ?

Ans. Two standard temperature points i.e. $0^\circ C$ and $100^\circ C$ are first fixed for temperature measurements. For various values of the sensing component resistance R as a function of the temperature T (in $^\circ$C), an equation or table is prepared. The equation is used when changes are linearly related to the variation in the physical environment. The table is used when changes are non-linearly related to the variation in the physical environment. A simple electronic circuit for measurement of temperature of oil or coolant plate in an IoT device using resistance bridge is shown in fig. 2.3. It uses the sensing component at the sensed object. At any measuring instance, the associated computing device calculates the temperature. The resistance bridge consists of a sensing resistor at the sensing object and three fixed or standard resistors. The microcontroller with the help of a serial port is connected to the sub-circuits, viz. serial port interface, analog-to-digital converter (ADC), signal conditioning amplifier, resistance bridge, and sensor resistance outputs A and B.
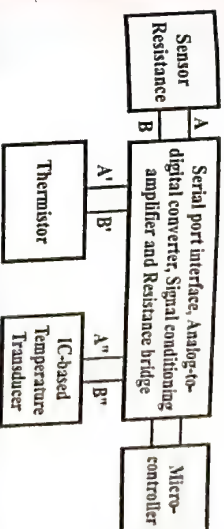

Fig. 2.3 Temperature Calculation using Resistance Sensor

In an alternative arrangement, circuit is connected to a thermistor output at A' and B' or IC-based temperature transducer at A" and B". The IC-based temperature transducer induces current in the output according to heat energy represented by temperature.

The output of resistance bridge circuit, act as input to the microcontroller, which after appropriate computation communicates the output.

Q.11. Discuss the working of a capacitive sensor.

Ans. In fig. 2.4, a circuit using a capacitance bridge is shown, which is made of a sensing capacitor (object) and three fixed capacitors. The microcontroller electronic circuit with port is connected to sub-circuits viz., serial port interface, analog-to-digital converter, signal conditioning amplifier, capacitance bridge and diode.
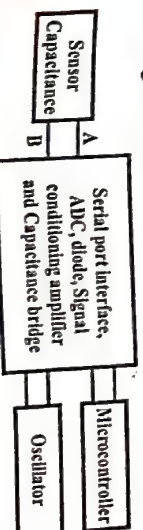

Fig. 2.4 Circuit for Capacitive Sensor

Let this sensor be used to check proximity of a metal object. The capacitance C of the sensing capacitor varies with the distance. This variation in C is converted to digital form by an ADC, which is then inputted to the microcontroller. The microcontroller after computation, communicates this data.

Q.12. Discuss the use of sensors for measurement of various parameters.

Ans. **Temperature** – The most commonly used sensor for temperature measurement is a thermistor which shows greater changes in resistance within narrow environment temperature range. A thermistor which shows negative temperature coefficient i.e. with rise in temperature value a drop in the resistance is known as NTC. It is used in home automation or in sensing the clouds. The output of a thermistor, is fed to the serial port of microcontroller through a circuit of SC amplifier and ADC.

A thermistor for which resistance rises with the rise in temperature, i.e. exhibits a positive temperature coefficient is known as PTC thermistor. Thin wires of platinum or other metallic alloys show this characteristic. PTC thermistors can be used for sensing temperature and measuring values over a wide range of temperatures.

The principle of a temperature sensor using a resistance bridge, signal conditioning amplifier and ADC for generating input to a microcontroller is discussed in Q.10.

Some ICs, e.g. AD590, which generates 1 μA for every 1°C rise in temperature, can be used as a temperature transducers. The circuit for such transducers is similar as shown in fig. 2.3, in which a resistance bridge is replaced by an IC transducer.

Most of the temperature sensors have three terminals – one for ADC input, V_{in} and two + and – V terminals for 5 V (or 3.6 V) supply.

Humidity – Humidity or relative humidity (RH) is measured in percentage. It is the relative percentage ratio of content of water vapour in air compared to the maximum possible water vapour content for the air temperature at the time of measurement. Percentage of relative humidity changes are measured using a capacitor sensor in the form of change in capacitance. The circuit shown in fig. 2.4, in which a capacitance bridge is replaced by a humidity sensor can be used for measurement of humidity.

Most of the humidity sensors show output voltage proportional to RH%. The output of these sensors, V_{RH}, which is a function of RH% is directly given to ADC, and ADC output is given to the serial port interface at microcontroller for measurement of RH%.

Distance – For measurement of distance of objects lying in the range of 0.15 m to 0.8 m infrared (IR) sensors are useful. In these sensors a narrow beam IR LED sends radiation at an inclined angle towards the object and reflected radiation is received by a nearby phototransistor (FPT). The time

delay between transmitted and reflected signals, is proportional to the object distance. The sensor LED is given input supply at + and – potential terminals and IR-FPT circuitry generates output V_{dis} as a function of distance. V_{dis} output is given as input to the ADC and ADC output is given to the input of microcontroller serial port to compute the distance. IR sensors can only be used for objects lying in the range of 0.15 to 0.8 m. Because above 0.8 m, the intensity of reflector radiation be come insufficient to detect and below 0.15 m, the time delay is less than 1 ns, which cannot be measured. For long range distances and any obstacles nearby ultrasonic sensors can be used. The ultrasonic waves have frequencies of the order of few kilocycles, and wave delay is 2 × 3 millisecond/m in air. Ultrasonic sensors found application in industrial automation and detection of rail tracks and oil pipeline faults.

Light – To detect light in the surrounding place, a photoconductor can be used. The resistance of the photoconductor drops in the presence of light, while its conductivity (reciprocal of resistance) increases exponentially with the increase in received light intensity. To measure incoming radiation intensity from a particular direction, a p-n junction photodiode can be used. The output of the sensor fed to the microcontroller through a SC amplifier and ADC.

Acceleration – Commonly used sensor for detection of linear accelerations a_x, a_y and a_z along three axes x, y, z is a micro-electro-mechanical sensor (MEMS). When a mass moves in a direction, an MEMS moves. The mechanical movement of MEMS causes variation in three capacitance values, C_x, C_y and C_z. The space between two plane surfaces which varies on acceleration along an axis, determines the value of each C. Variations in the capacitances cause variation in the voltages of an electronic circuit, which gives measure of a_x, a_y and a_z. Modern smart phones uses an accelerometer sensor, which rotates with the phone.

Using this sensor, acceleration components in three directions can be detected and on the basis of that display screen image and menu items can be rotated and aligned horizontally on vertically. An accelerometer can also detect up/down, left/right and front/ back accelerations given by the user during playing games on the phone. An accelerometer circuit is shown in fig. 2.5.
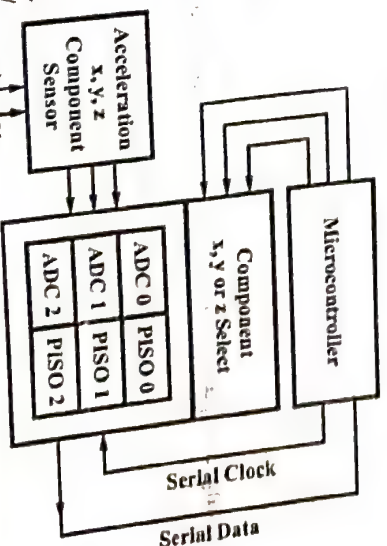

Fig. 2.5 Measuring Three Acceleration Components using Accelerometer Sensor

Vibrations and Shocks – An MEMS using piezoelectric effect in place of capacitive change effects can be used for detection of vibrations and shocks. Vibrations cause repeated compression and decompression of piezoelectric materials. Depending upon the intensity of vibrations, an associated electronic circuit generates output. The mechanical shocks can also be sensed by this sensor. The in-built sensor in the mobile senses changes due to user initiated vibrations or shocks or due to accidental fall of device and the operating system of mobile takes action accordingly.

Q.13. What are the uses of an analog sensor? What are the uses of a digital sensors?

(R.G.P.V., May 2018)

Ans. Sensors are of two types, analog and digital. Sensors can sense temperature, humidity, acceleration, angular acceleration, object distance, magnetic object proximity, motion, sound etc.

The uses of digital sensor are as, sense the number of chocolates of each type of flavours remaining unsold and communicate that number to chocolate fill service and sense unoccupied parking slots.

Also refer Q.12.

Q.14. Explain in brief about –
(i) *Serial port interface* (ii) *Analog-to-digital converter*
(iii) *Sampling ADC* (iv) *Signal conditioning amplifier.*

Ans. (i) *Serial Port Interface* – ADC 8 or 10 or 12 bit output is input to the interface which sends input to the serial port of microcontroller. The output of serial port interface has only two terminals.

(ii) *Analog-to-digital Converter* – The analog output of a signal conditioning amplifier is inputted to the ADC as V_{in}. The decimal value of digital output of ADC related to V_{in} is in the form of binary bits at the input. The decimal value of binary bits is proportional to the ratio of analog input voltage V_{in} and reference voltage V_{ref}.

The output of ADC is inputted to the microcontroller. Alternatively a microcontroller may consists of an in-circuit ADC or multiple inputs ADC. The digital output of the in-circuit ADC is processed by the microcontroller.

For example, the digital output of an 8 bits $[2^n - 1 = 255_d]$ ADC will be

Since V_{ref} is the maximum input which can be applied to the ADC, therefore maximum digital output will be 11111111_b, i.e. 255_d,

$$= V_{in} \times 255_d / V_{ref}$$

(iii) *Sampling ADC* – When ADC accepts input signals at fixed periodic intervals and converts them into digital outputs, the process is known as sampling ADC. In sampling the interval is set on the basis of signal frequency

and other requirements. Sampling ADC finds many applications, e.g. during music or voice or song recording, the sampling ADC receives signals from the microphone for the recording sensor.

(iv) *Signal Conditioning Amplifier* – An SC amplifier, amplifies the input signal, and conditions the minimum $[V_{in} (min)]$ and maximum $[V_{in} (max)]$ values of the sensed physical parameter, so that they become equal to 0V and V_{ref} respectively of the output of SC.

Q.15. Give use of scalar sensors.

Ans. Scalar sensors produce output signal or voltage which is generally proportional to the magnitude of the quantity being measured. Physical quantities such as temperature, colour, pressure, strain, etc. are all scalar quantities as only their magnitude is sufficient to convey an information.

For example, the temperature of a room can be measured using a thermometer or thermocouple, which responds to temperature changes irrespective of the orientation of the sensor or its direction.

Q.16. Give use of vector sensors.

Ans. Vector sensors produce output signal or voltage which is generally proportional to the magnitude, direction, as well as the orientation of the quantity being measured. Physical quantities such as sound, image, velocity, acceleration, orientation, etc. are all vector quantities, as only their magnitude is not sufficient to convey the complete information.

For example, the acceleration of a body can be measured using an accelerometer, which gives the components of acceleration of the body with respect to the x, y, z coordinate axes.

Q.17. Name commonly used protocols for sensor data communication.

Ans. Sensor data communication is done through serial interfaces. Wired communication can be of following two types –

(i) Asynchronous serial communication like using UART communication. A 125 kHz RFID UART module is used by a RFID reader. A GPS device also sends serial data by using the UART module.

(ii) Synchronous serial communication, used by the sensors to communicate serial data using I2C or SPI interfaces.

Commonly used communication protocols for serial bus communication are Inter-Integrated Circuit (I2C), Universal Serial Bus (USB) and FireWire (or IEEE 1394). Some other protocols which are used for serial communication in automobiles' internal sensors and other embedded circuits are LIN (Local Interconnect Network), CAN (Controller Area Netwok) and MOST (Media Oriented System Transport).

Q.18. Discuss various protocols used for sensor communication.

Ans. Various protocols used for sensor communication are discussed below –

(i) *UART for Serial Bus* – A UART device transmits 8-bit data at successive intervals, known as baud intervals. Baud is a German word, which means drop, just like a raindrop. In UART protocol bits are communicated like bauds. Each character, which communicates 8-bit is coded as per the ASCII code. The 8-bit data is preceded by a start bit (where 1 becomes 0 for a certain interval, i.e. baud interval), and succeeded by a stop bit (which means 1 for minimum interval, viz. baud interval). Thus, each character contains 10 bauds.

(ii) *Using UART Communication for a RFID Tag* – A library program for RFID using UART serial interface protocol enables the use of UART communication for an RFID tag.

A tag ID has ten digit characters, which is preceded by a header character and followed by an end character which consists of 1 byte. The total number of digits communicated for each tag is equal to 12.

(iii) *Using I2C Protocol for a Serial Bus* – In an I2C (Inter-Integrated Circuit) bus, different integrated circuits using I2C interface communicate over the same set of wires. Using I2C protocol, number of devices can be interconnected for serial bus communication, such as one for flash memory, one for touch screen, and one for measuring pressure in a number of processes in a plant. The sensors using I2C protocol consists two active signal terminals, viz. Serial Data (SDA) and Serial Clock (SCL) lines and one ground terminal. An example of serial synchronous data communication using I2C protocol is shown in fig. 2.6. A device which sends signals using SCL and SDA lines is called master and the device receiving these signals is called slave.
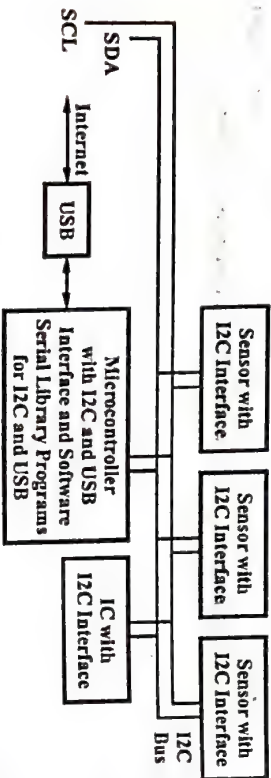

Fig. 2.6

(iv) *Using USB Bus* – A bus which connects a host system and a number of interconnected peripheral devices is known as universal serial bus

(USB). As much as 127 devices can be connected to a host. Use of USB bus for serial communication is enabled by a library program for USB serial interface protocol. There are three standard protocols, which are used for USB protocol, viz. USB 1.1 (a low speed of 1.5 Mbps, 3 m channel and a high speed 12 Mbps, 25 m channel); USB 2.0 (high speed of 480 Mbps 3 m channel); wireless USB (high speed 480 Mbps 25 m channel). By using a USB port driving software and a host controller, a host connects to the devices or nodes. Tree topology is used by USB bus. The host computer has a host-controller, which connects to a root hub. A hub is connected to other nodes or hubs.

(v) *Using the IEEE 1394 Bus Standard Protocol* – IEEE 1394 is a high speed standard, which is used for 800 Mbps serial isosynchronous data transfer, in which bits in data frame communicate synchronously but frames in-between time interval can be variable. This type of high speed bus interface is required by latest hard disk drives, printers, high-definition audio-video and music systems multimedia peripheral, digital video cameras, digital camcorders, digital video disk, set-top boxes, to directly connect to a personal computer.

(vi) *Using the LIN Serial Bus* – Local Interconnect Network (LIN) is a serial bus network with single master and upto 15 slaves, without bus arbitration. Without arbitration means, when number of devices request to use the bus, the destination is not selected by the bus. LIN protocol is simpler to use compared to CAN in automobiles. It can be used for communication between automobile circuits, sensors and actuator circuits, and components and systems.

Some special features of LIN are given below –

(a) Length of data frame can be varied.

(b) Using single wire communication upto 19.2 kbit/s for 40 m bus length can be done.

(c) Flexibility in configuration.

(vii) *Using the CAN Protocol for Serial Bus* – A serial bidirectional line network of multiple CAN controllers and devices is shown in fig. 2.7. A
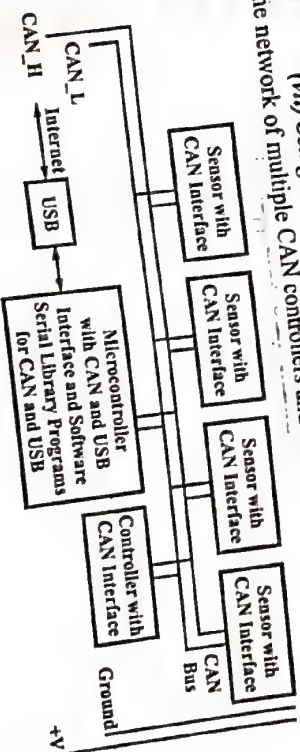

Fig. 2.7

CAN bus is a standard bus used in a distributed network, such as in a Vehicular Control Network (VCN), where a number of devices (e.g. controller for brakes, engine, power, lamps, gates, windows, air conditioning, front display panel, meter display panels and cruising) are located and distributed.

CAN bus has a serial bidirectional line. A CAN device can operate at a maximum rate of 1 Mbps and receives or sends one bit at an instance. A twisted pair connection is employed at each node, which can run upto a length of 40 m.

Using a CAN bus, number of different embedded controllers with sensors and actuators can be networked and controlled.

(viii) Using the MOST Protocol – Use of MOST protocol enables high speed serial bus for synchronous data communication, required to form a multimedia network used in automotive IoT and IIoT applications. Communication using MOST protocol between a MOST Network Interface Controller (NIC) and devices is facilitated by a driver software called MOST network services. With the help of these services a user can communicate media files.

Q.19. Discuss the role of sensors in IoT. Explain various types of sensors.

(R.G.P.V., May 2019)

Ans. A large varieties and different types of sensors are available in the market at present. They are used for the improvement of quality of human life style. The importance of IoT rises day by day, a sensor in its part is designed for the measurement of physical external stimulus and records, indicates or responds to it that can be read by a user or another device. The most commonly used sensors in Internet of things are described below –

(i) Temperature Sensors – These are one of the commonly used sensors that measure the temperature or heat of a given medium. These sensors use a number of methods to determine and quantify the temperature of any object. Some of the temperature sensors required a physical contact with the object while other types do not require contact as they can detect liquid or gases that emit radiant energy like spike in heat or temperature. Highly sensitive semiconductors available in market which are capable enough to monitor and display slight variation in temperature.

(ii) Proximity Sensors – Proximity sensors are the best to detect any type of motion. They are widely used in applications such as security, safety, or efficiency. These sensors are used to avoid obstacles in navigating to a crowded place or any complex route as best possible sensor for the map building. Proximity sensors use electromagnetic radiation like radar signals to detect motion or habitation. Proximity sensors have various industrial uses.

The retailers use proximity sensors to find out vicinity of the customers once they are near to their premises by sending them some offers on their IoT devices. It also can be used in parking systems, museums, airports, etc.

(iii) Pressure Sensors – Pressure sensors are used for measuring pressure of any type of gas or liquid. Pressure sensors convert the physical power into an electrical signal. They also can be effectively used for measuring other variables like speed and altitude or similar situation in some way. Barometers and pressure gauges are the common using pressure sensors used for IoT system. Barometers are helpful in weather forecasting as it can give accurate measurement of ambient air. Pressure gauges are mostly used in industrial sites as it is good for the monitoring of pressure in closed environments. Pressure sensors are ultimate solution for IoT devices as these can be used for various areas such as touch screen devices, bio medical devices, automotive systems and manufacturing industry. Micro pressure sensors are type of small size sensors for the measurement of pressure. The first micro sensor was developed and used by industry is piezoresistive pressure sensor to reduce fuel consumption by maintaining a tight control ratio between air and fuel and other is disposable blood-pressure sensor to monitor the corresponding status of the patient during operation. The market available products are usually either piezoresistive or capacitive. Micro pressure sensors work on the principle of mechanical bending of thin silicon diaphragm by the contact air or gas pressure. This physical movement is converted into electrical output.

(iv) Optical Sensors – Optic sensing technology is used to detect electromagnetic energies like light. It utilizes concept of the photoelectric effect, such as there will be an ejection of electrons, when a negatively charged plate of some appropriate light-sensitive material, is striked by a beam of photons. The electrons can then be made to flow as a current from the plate feed as a signal. The magnitude of the electric current produced is directly proportional to the light intensity or number of photons. They can emit, receive, and convert light energy into electrical signal. The fiber optic sensor IoT interface is connected to Internet and can collect various information for monitoring different parameters. These optical sensors are widely used in different types of digital cameras which act as one of the major physical devices of an IoT system. As they are passive to all forms of electrical interfaces, they are preferred sensors for IoT. Optical sensors are very good for energy, health care, aerospace, chemicals, environmental IoT systems. Optical sensors can be ideal for environments such as oil refineries, mining operations, pharmaceutical companies, and chemical industries due to its no risk components.

(v) Humidity Sensor – Humidity is the presence of water in air. The amount of water vapour in air can affect human livings as well as many

manufacturing processes. The presence of water vapour also influences various physical, chemical, and biological activities and its measurement in industries is critical because it may affect the quality and cost of the product, the health and safety of the personnel. So humidity sensing is important in the control systems for industrial processes and human beings. Controlling or monitoring of humidity is very important in many industrial, agricultural and domestic applications. In semiconductor manufacturing process, the humidity or moisture levels needs to be properly controlled and monitored while wafer processing. In medical science humidity control is required for respiratory supporting system, sterilizers, incubators, pharmaceutical processing and many other biological products. Humidity sensor is also necessary in chemical gas purification, dryers, ovens, paper, textile production and food processing industry. In agriculture, measurement of humidity is important for plant protection, such as dew control, soil moisture testing and monitoring. Domestic applications requires humidity control for living environment in buildings and for microwave ovens, etc.

Types of Sensors – Basically two types of sensors are as follows –

　(i) Physical sensor　(ii) Chemical sensor.

　(i)　Physical Sensor – Physical sensors are generally used for measuring physical quantities such as length, temperature, pressure, electricity, weight, sound, etc. It can be defined as a device which respond to physical property, called stimulus, and produce a corresponding measurable electrical signal.

　(ii)　Chemical Sensor – A device which responds to a particular way by a chemical reaction and that can be used for the quantitative or qualitative determination of the change is called chemical sensor. Such sensor is concerned with detecting and measuring a specific chemical substance or set of chemicals.

Q.20. What types of sensors are used in a mobile phone ?

　Ans. With the change in physical conditions, a characteristic parameter of a circuit changes. The technology, which facilitates such changes due to sensing is used in mobiles. A mobile phone can sense surrounding conditions using proximity sensors. A finger touch and gestures can be sensed by touchscreen of a mobile phone. A whole lot of sensors including resistive and capacitive sensors, photodiode current based sensors, acceleration, gyroscope, temperature and pressure sensors are used in smartphones. The functioning of various applications and games is also enabled by sensors. A microcontroller associated with a sensor circuit calculates the touched position and maps it to a user command in a resistive touchscreen, on the basis of which the mobile takes further actions.

Q.21. Explain the concept of participatory sensing.

　Ans. Participatory sensing refers to the sensing by the individuals and group of people who are contributing sensory information to form a body of knowledge. Deborah Estrin defines participatory sensing as, "a process whereby individuals and communities use increasingly capable mobile phones and cloud services to collect and analyse systematic data for use in discovery.
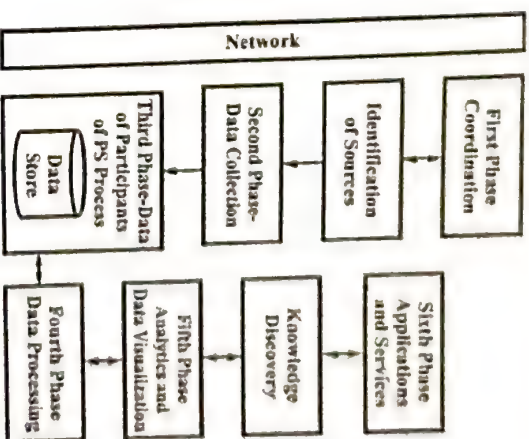

Fig. 2.8 Participatory Sensing Process

Various sensors used in mobile phones e.g., camera, temperature and humidity sensors, an accelerometer a gyroscope, a compass, infrared sensors, NFC sensors, QR code readers, microphone and GPS are participants of a PS process. The sensed information can be communicated on the Internet.

　Various sources of data in a PS process can be classified in three categories –

　(i)　Individual Data Collectors – They communicate data regarding weather, waste collection, etc.

　(ii)　Group Data Collectors – They include traffic lights, which can send data about traffic density and parking availability. Smart automobiles can also communicate traffic density data at different locations.

　(iii)　Social Sites Data – They include Facebook, Twitter, LinkedIn, etc.

　Various phases of a PS process used for IoT applications is shown in fig.

2.8. First phase is the coordination phase, in which various participants coordinate with each other after identifying the sources. Second and third

phases involve data collection, communication and its storage on cloud servers. In fourth phase data so stored is processed, which leads to analytics and data visualization in fifth phase and finally results in knowledge discovery. Using this knowledge appropriate actions are initiated in sixth phase.

Participatory sensing is used for retrieving information about weather, environment, pollution, waste management, traffic congestion, urban mobility or disaster management.

Various challenges related to participatory sensing are security, privacy, reputation and ineffective incentives to the participants.

Q.22. Explain the following measurement terms –

(i) *Bias*

(ii) *Drift*

(iii) *Hysteresis*

(iv) *Repeatability*

(v) *Stability.*

Ans. **(i)** *Bias* – Bias is consistent, repeatable errors in the set of measurements. This error is defined as the average of measured values minus the true value.

(ii) *Drift* – This is the low frequency change in a sensor with time. It is often associated with electronic aging of components or reference standards in the sensor. Drift generally decreases with the age of a sensor as the component parts mature. A smoothly drifting sensor can be corrected for drift, e.g. Sea Bird temperature sensors that are drifting about 1 m°C/yr (and have been smoothly changing for several years) allow one to correct for the drift and get more accurate readings. Drift is also caused by biofouling that can not be properly corrected for, but we often try.

(iii) *Hysteresis* – A linear up and down input to a sensor, results in an output that lags the input, e.g. we get one curve on increasing pressure and another on decreasing. Many pressure sensors have this problem, for better ones it can be ignored. It is often seen in a CTD when the pressure reading on deck after recovery is different from the reading before it is deployed. It is not a problem with the response time of the sensor, but is an inherent property of some sensors that is undesirable. In a CTD it also may be a temperature sensitivity problem.

(iv) *Repeatability* – This is the ability of a sensor to repeat a measurement when put back in the same environment. It is often directly related to accuracy, but a sensor can be inaccurate, yet be repeatable in making observations.

(v) *Stability* – This is another way of stating drift. For a given input we should always get the same output. Drift, short and long term stability are actually ways of expressing a sensor's noise as a function of frequency. Sometimes this is expressed as guaranteed accuracy over a certain time period. Drift is often a problem with pressure sensors under high pressure. All sensors drift with time – hence the standardization of PRTs in triple-point-of-water and gallium melt cells.

Q.23. Discuss various types of sensor errors.

Ans. Various types of sensor errors are discussed below –

(i) *Bias (or Systematic) Errors* – For definition of bias error, refer Q.22.

A nondimensional form of bias error is the mean bias error, defined as –

MBE = systematic error/true value. Bias errors arise for many reasons such as –

(a) *Calibration Errors* – Perhaps due to nonlinearity or errors in the calibration method.

(b) *Loading or Intrusion Errors* – The sensor may actually change the very thing it is trying to measure.

(c) *Spatial Errors* – These errors arise when a quantity varies in space, but a measurement is taken only at one location (e.g. temperature in a room; usually the top of a room is warmer than the bottom).

(d) *Human Errors* – These can arise if a person consistently reads a scale on the low side, for example.

(e) *Defective Equipment Errors* – These arise if the instrument consistently reads too high or too low due to some internal problem or damage.

(ii) *Drift Error* – The output changes (drifts) from its correct value, even though the input remains constant. Drift error can often be seen in the zero reading, which may fluctuate randomly due to electrical noise and other random causes, or it can drift higher or lower (zero drift) due to nonrandom causes, such as a slow increase in air temperature in the room. Thus, drift error can be either random or systematic.

(iii) *Hysteresis Error* – The output of a system can be different, depending on whether the input is increasing or decreasing at the time of measurement. This is a separate error from instrument repeatability error. For example, a motor-driven traverse may fall short of its reading due to friction, and the effect would be of opposite sign when the traverse arrives at the same point from the opposite direction, thus, hysteresis error is a systematic error, not a random error.

(iv) Quantization Error – When the sensor has a digital output, the output is essentially an approximation of the measured property. This error is called quantization error.

(v) Accuracy – Accuracy is the closeness of agreement between a measured value and the true value. Accuracy error is formally defined as the measured value minus the true value. The accuracy error of a reading (which may also be called inaccuracy or uncertainty) represents a combination of bias and precision errors. The overall accuracy error (or the overall inaccuracy) of a set of readings is defined as the average of all readings minus the true value. Thus, overall accuracy error is identical to systematic or bias error.

(vi) Precision – Precision characterizes the random error of the instrument's output. Precision error (of one reading) is defined as the reading minus the average of readings. Thus, precision error is identical to random error. The fig. 2.9 shows the difference between precision and accuracy. Two people, A and B, shoot guns at targets. Both people shoot eight times. Each plus sign marks the spot where a bullet hits the target. We can say – A is more precise than B, but not as accurate. On the other hand, B is more accurate than A, but not as precise. Instrument precision is often associated with instrument resolution, but these are not the same thing. An instrument can have great resolution, but poor precision.

Fig. 2.9

(vii) Zero Error – The instrument does not read zero when the input is zero. Zero error is a type of bias error that offsets all measurements taken by the instrument, but can usually be corrected by some kind of zero offset adjustment. Zero balance is a term used by manufacturers to indicate the maximum expected zero error of their instrument.

(viii) Linearity Error – The output deviates from the calibrated linear relationship between the input and the output. Linearity error is a type of bias error, but unlike zero error, the degree of error varies with the magnitude of the reading.

calibrated exactly in the first place. Since this affects all readings by the instrument, this is a type of systematic or bias error.

(ix) Sensitivity Error – The slope of the output vs. input curve is not calibrated exactly in the first place. Since this affects all readings by the instrument, this is a type of systematic or bias error.

(x) Resolution Error – The output precision is limited to discrete steps (e.g., if one reads to the nearest millimeter on a ruler, the resolution is around +/– 1 mm). Resolution error is a type of random error.

(xi) Instrument Repeatability Error – The instrument gives a different output, when the input returns to the same value, and the procedure to get to that value is the same. The reasons for the differences are usually random, so instrument repeatability error is a type of random error.

ACTUATOR, ACTUATOR TYPES – HYDRAULIC, PNEUMATIC, ELECTRICAL, THERMAL/MAGNETIC, MECHANICAL ACTUATORS, SOFT ACTUATORS

Q.24. What is an actuator? Discuss some examples involving application of actuators.

Or

Write short note on actuator.

(R.G.P.V., May 2018)

Ans. A device which takes the input in the form of command, pulse or state (1 or 0) or set of states (1s or 0s) or a control signal and performs some function is called an actuator. Examples of applications of actuators are given below –

  (i)   Light sources
  (ii)   LEDs
  (iii) Piezoelectric vibrators and sounders
  (iv) Ringing of alarm bell
  (v)   Applying brakes on a moving vehicle
  (vi) Switching on a set of streetlights
  (vii) Relay switch
  (viii) Speakers
  (ix) Solenoids
  (x)   Servomotor
  (xi) Switching on or off a heater or air-conditioner or boiler current in a steam boiler in a thermal plant.

Some of these applications are discussed below –

  (i) LED – LED is an actuator which accepts pulse width modulated (PWM) pulses, generated by a microcontroller, and emits light or infrared

radiation. By controlling the inputs, we can use different colour LEDs, RGB LEDs, vary the colours and intensity of LED and display graphic and text on big screen.

(ii) Piezoelectric Vibrator – When varying electric voltages are applied at the input of piezoelectric crystals, these generate vibrations.

(iii) Piezoelectric Speaker – Synthesized music tunes and sounds can be created using a piezoelectric speaker. When appropriately programmed pulses are inputted to the speaker they generate the music, sounds, buzzers and alarms.

(iv) Relay Switch – A relay switch is an electronic switch which makes mechanical contact when the input circuit is magnetised with a control input 1 or 0 from the port pin of a microcontroller or by a push button switch and battery. Depending upon the state 1 or 0, the current flows through the switch or voltage is applied across it.

(v) Solenoid – A solenoid is a special type of actuator having a number of cylindrically wound coils. A magnetic field is created, when current flows through the coils. The intensity of magnetic field depends upon the number of turns in the solenoid and the magnitude of current in it. Now, if a iron shaft is placed along the axis of the solenoid, its movement can be controlled by varying the input current and pulses of current. The shaft can be made to forward or backward or to have a repeated to and fro motion. Using a cam mechanism, the linear motion of shaft can be converted into the rotary motion.

(vi) Motor – A motor can be AC or DC controlled. IO modules can be used with these motors to deliver high currents on receiving the control digital inputs of 1s and 0s. The motor is rotated by AC or DC. Rotary motion of a motor can be converted into linear motion using a cam.

(vii) Servomotor – A servomotor is a geared DC motor, which is commonly used in robotic applications to rotate the motor shaft. The servomotor has 3 terminals, two for voltage supply (+ and – terminals) and one for a pulse width modulated (PWM) input. Depending upon the value of PWM input, the microcontroller generates pulse of different widths.

Q.25. Explain the various types of actuator.

Ans. The various types of actuator are as follows.

(i) Hydraulic Actuators – These actuators are simple devices with mechanical parts that are used on linear or quarter-turn valves. They are designed based on Pascal's law, viz. when there is an increase in pressure at any point in a confined incompressible fluid, then there is an equal increase at every

point in the container. Hydraulic actuators comprise of a cylinder or fluid motor that utilizes hydraulic power to enable a mechanical process. The mechanical motion gives an output in terms of linear, rotary, or oscillatory motion. Hydraulic actuators can be operated through a hydraulic pump, which can be operated manually, such as a hydraulic car jack, or they can be operated manually, such as a hydraulic car construction equipment such as cranes or excavators. The disadvantage with the hydraulic actuators is that they are more complex and need more maintenance.

(ii) Pneumatic Actuators – These actuators work on the same concept as hydraulic actuators except that the compressed gas is used instead of liquid. Pneumatic actuators use air under pressure that is most suitable for low to medium force, short stroke, and high-speed applications.

(iii) Manual Actuators and Automatic Actuators – Manual actuators employ levers, gears, or wheels to enable movement, while an automatic actuator has an external power source to provide motion to operate a valve automatically. Power actuators are a necessity on valves in pipelines located in remote areas.

(iv) Electrical Actuators – Electrical actuator is an electromechanical device that converts electrical energy into mechanical energy. Most electric actuators operate through the interaction of magnetic fields and current-carrying conductors to generate force. The reverse process, producing electrical energy from mechanical energy, is done by generators such as an alternator or a dynamo; some electric actuators can also be used as generators, for example, a traction motor on a vehicle may perform both tasks. Electric actuators and generators are commonly referred to as electric machines.

Electric actuators are found in applications as diverse as industrial fans, blowers and pumps, machine tools, household appliances, power tools, and disk drives. They may be powered by direct current, e.g., a battery powered portable device or motor vehicle, or by alternating current from a central electrical distribution grid or inverter. Small actuators may be found in electric wristwatches. Medium-size motors of highly standardized dimensions and characteristics provide convenient mechanical power for industrial uses.

(v) Thermal/Magnetic Actuator – One type of thermal actuator is a bimetallic strip. This device directly converts thermal energy into motion. This is accomplished by utilizing an effect called thermal expansion.

Thermal expansion is the manifestation of a change in thermal energy in a material. When a material is heated, the average distance between atoms (or molecules) increases. The amount of distance differs for different types of material. This microscopic increase in distance is unperceivable to the human

eye. However, because of the huge numbers of atoms (or molecules) in a piece of material, the material expands considerably and, at times, is noticeable to the human eye.

(vi) *Mechanical Actuators* – Mechanical actuators convert a mechanical input into linear or rotary motion. A common example of a mechanical actuator is a screw jack. The picture shows a screw jack in operation. Rotation of the screw causes the legs of the jack to move apart or move together. Inspecting the motion of the top point of the jack, this mechanical rotational input is converted into linear mechanical motion.

Mechanical actuators can produce a rotational output as well with the proper gearing mechanism.
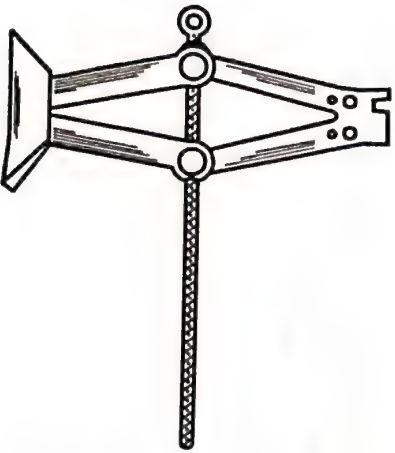

Fig. 2.10

Q.26. *Define electronic sensors and actuator in detail.*

(R.G.P.V., Nov. 2019)

Ans. Refer to Q.2 and Q.24.

●●

BASICS OF IoT NETWORKING, IoT COMPONENTS, FUNCTIONAL COMPONENTS OF IoT, IoT SERVICE ORIENTED ARCHITECTURE, IoT CHALLENGES

Q.1. Discuss the basic concept of IoT networking.

Ans. In an IoT network, some of the major technologies are wireless personal area network viz. 6LoWPAN, ZigBee, Bluetooth. In addition, on a slightly larger wireless network scale, Wi-Fi, wireless LAN technology, is going to be used and needs to be supported. Then as, on the larger scale network, the backbone, is used in addition to the mobile communication network domain. Then smartphones and mobile communication phones will also be used and they will be connected to the base stations. The base stations will provide connectivity to the wider network, the internet. Considering this, many options are needed to think about. One thing is that smartphones are equipped with Bluetooth and Wi-Fi therefore IoT network is possible. The most simple topology control will be a wireless PAN which is either like a Bluetooth or some other type of network. That is connected to a smartphone, and the smartphone will bring that signal up and connect it through LTE, or 4G/3G to the base station, and the base station will connect to that wide array network, which is the Internet. It can be observed that one technology linking on a technology of something.

Wearable devices such as shoes, watch, glasses, belt, etc. can be used to detect biometric information because they are close and attached to body and they can sense in real time. These devices can send data through Bluetooth technology to the smart phone that rests inside the pocket. In that case, the smartphone will be able to pick up this information and it will be able to use it through mobile communication link to send it to a base station. The base station, which is connected to a wide area network and the Internet, is going to be where the information connects to any other place we need in the world. For example, if a person is using smart watch and shoes or belt or something similar attached to his body, to monitor his health condition that data can be sent wirelessly to his smartphone then, the smartphone will send that information

to the base station, which will be connected to the internet and that information can be delivered to a control centre or medical server. If something happens to the person, they will know and they will be able to provide immediate support. A smart device can be used to collect the information that communicates with control centres or like a medical centre and this is how everything gets connected together.
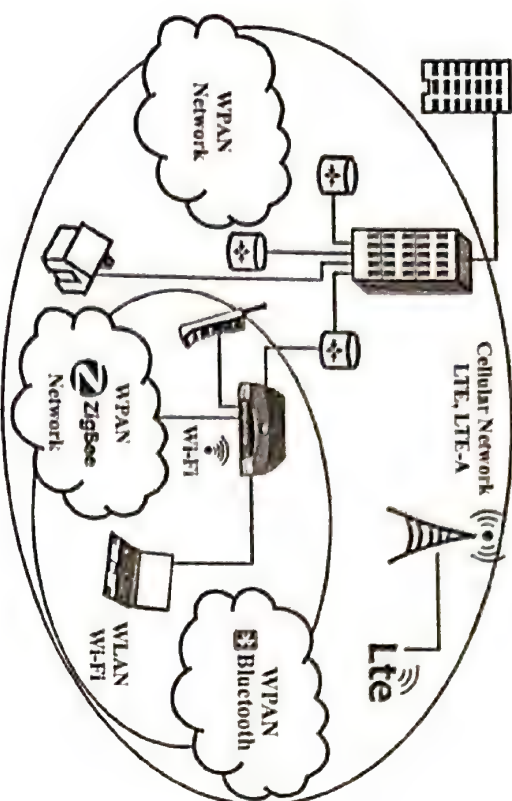

Fig. 3.1 IoT Networking

Q.2. What are the components of IoT ?

Ans. Components of IoT are as follows –

(i) Hardware – It consists of a firmware, control unit, microcontroller, communication module, sensors and actuators.

(ii) Software – In IoT, software is required for actions on messages, information and commands which the devices receive and then output to the actuators, which enable actions such as robotic hand movement.

(iii) Physical Object – It refers to a hardware with embedded software.

(iv) Communication Module – These include software consisting of device APIs and device interface for communication over the network and communication circuit/port(s), and middleware for creating communication stacks using 6LowPAN, CoAP, LWM2M, IPv4, IPv6 and other protocols.

Q.3. What are the functional components of IoT ?

Ans. Functional components of IoT are as follows –

(i) Component for interaction and communication with other IoT devices.

(ii) Component for processing and analysis of operations.

(iii) Component for Internet interaction.

(iv) Components for handling Web services of applications.

(v) Component to integrate application services.

(vi) User interface to access IoT.

Q.4. Discuss about the IoT service oriented architecture.

Ans. The IoT service oriented architecture is shown in fig. 3.2 In IoT we have four different layers. Viz. the sensing layer, the network layer, the service layer and the interface layer. Sensing layer basically takes care of sensing through different RFID tags, sensors and so on and, data sensed or acquired are sent to the next layer higher up which is the network layer. The network layer basically serves sensor networks, social networks, different other networks and databases, Internet and so on. Then we have the service layer
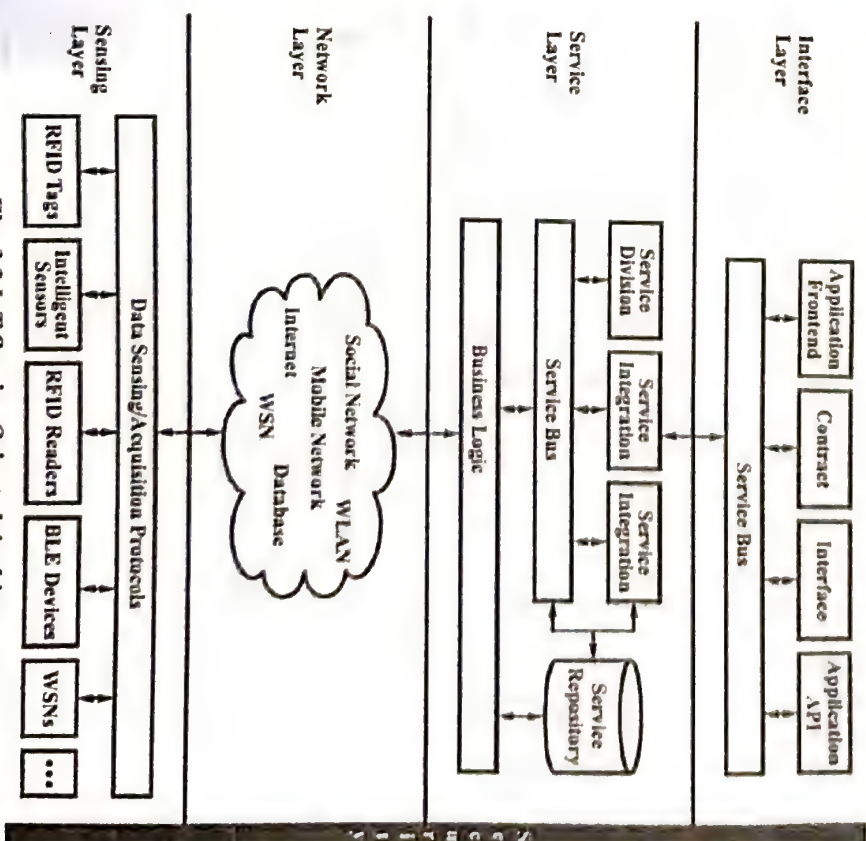

Fig. 3.2 IoT Service Oriented Architecture

which deals mostly with the service delivery such as service division, service integration, service repository, service logic, business logic and so on. So, all these different things have evolved with the offering of the services to support the different business functions. Then, we have the interface layer, on which we have the application frontend, a contract, interface and application APIs. The security issues span across all these layers.

Q.5. What are the challenges of IoT ? Explain.

Ans. The challenges of IoT are as follows –

(i) Availability – Availability of the IoT must be realized in the hardware and software levels to provide anywhere and anytime services for customers. Availability of software refers to the ability of the IoT applications to provide services for everyone at different places simultaneously.

(ii) Security Concerns – If the IoT devices are poorly secured, cyber attackers will use them as entry points to cause harm to other devices in the network. This will lead to loss of personal data out into the public.

(iii) Privacy Issues – These devices collect user data without their permission, analyze them for purposes only known to the parent company. The social embrace of the IoT devices leads people to trust these devices with collection of their personal data without understanding the future implications.

(iv) Inter-operers Ability Standard Issues – In an ideal environment, information exchange should take place between all the interconnected IoT devices. But the actual scenario is inherently more complex and depends on various levels of communication protocols stacks between such devices.

Q.6. Explain the network function virtualization (NFV).

Ans. NFV is a technology that leverages virtualization to consolidate the heterogeneous network devices onto industry standard high volume servers, switches and storage. NFV is complementary to SDN as NFV may give the
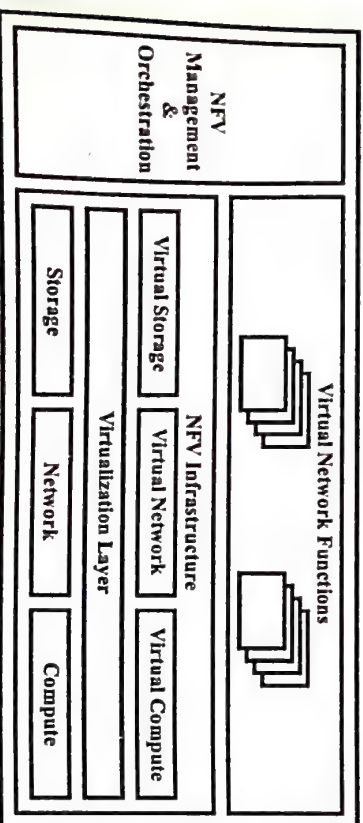

Fig. 3.3 NFV Architecture

infrastructure on which SDN can run. NFV and SDN is not dependent but both are mutually beneficial to each other. SDN can run without NFV and without SDN, network functions can be virtualized. The network function virtualization (NFV) is shown in fig. 3.3.

The main elements of the NFV architecture are as follows –

(i) Virtualized Network Function – It is software implementation of a network function which is able to run over the NFV infrastructure.

(ii) NFV Infrastructure – It involves virtualized compute, network and storage resources.

(iii) NFV Management and Orchestration – It concentrates on all virtualization-particular management operations. The life cycle management and orchestration of these software resources which support the infrastructure virtualization and life cycle management of VNFs is covered by it.

Q.7. Describe how NFV can be used for virtualizing IoT devices.

(R.G.P.V., May 2018)

Ans. In software, network function virtualization has a network functions implemented which run on virtualized resources in the cloud. Separation of network functions can be enabled by NFV that may be implemented in the software by underlying hardware. By installing new software, hence network functions may be simply tested and upgraded, whereas the hardware remains the same. Virtualizing network functions minimizes the power consumption, and also minimizes equipment costs. Virtualized network functions are permitted by the multi-tenanted nature of the cloud to be shared to multiple network services. In case of fixed and mobile networks, NFV is applicable only for control plane and data plane functions. Several functions are performed by home gateway including – network address translation (NAT), dynamic host configuration protocol (DHCP) server, application specific gateway and firewall. Private IP addresses is provided by home gateway to every attached device in the home. The home gateway gives routing capabilities and it converts the private IP address to one public address. For applications like IPTV and VoIP, application specific routing is provided by the gateway. Virtualized home gateway is hosting by NFV infrastructure in the cloud. The virtualized gateway provides private IP address to the each device in the home. The virtualized gateway connects to network services like VoIP and IPTV too.

Q.8. What is network function virtualization (NFV) based architecture to address connectivity and interoperability challenges in Internet of things ?

(R.G.P.V., May 2019)

Ans. Connectivity Challenges in IoT – Connectivity is the fundamental component of IoT because transport of data from one IoT device or system to another depends very much on whether they are able to connect with each

other or not. Difference in connectivity protocols can be studied at various layers of the communication stack. For example, physical layer, MAC layer, network layer and transport layer.

(i) Physical Layer – Deployment of IoT solutions depends largely on the availability and cost of the physical hardware that operates on the allowed frequency band in the country of deployment. Since the cost of IoT deployment depends largely on the mass production ability of a particular sensor/actuator, it is necessary to have some form of agreement on the allocated license frequency band for IoT.

(ii) MAC Layer – Presence of multiple vendor solutions and low-power communication technologies are the main cause of concern at the MAC layer. Different vendors have their preference on the MAC layer technologies for various reasons such as team capability, cost, ease of use etc. Various technology options at the data link layer include wired LAN, Wi-Fi, bluetooth, ZigBee, 6LoWPAN, proprietary Sub-1GHz protocols, power line communication (PLC), fieldbus and many others. Moreover, many of the MAC protocols have already been enhanced to cater for better battery life.

Most standardization bodies are spending excessive amount of time trying to address heterogeneity at the MAC layer. However, the implementation of some of these technologies spans over different layers of the OSI model and makes it difficult to come to a consensus. Furthermore, standards are simply recommendations and not certification so the vendors have a choice to comply, partially comply or not to comply with the standards.

(iii) Network Layer – In order to exchange information with each other, IoT devices need to be connected to the Internet or have a mechanism to push data to the Internet. Hence, they need to support the TCP/IP protocol suite that is too bulky and not cost effective for the IoT device.

One of the biggest challenges of using reduced TCP/IP in IoT is maximum transmission unit (MTU) size. IoT devices operate with much smaller MTU of about 127 bytes compared to computers, which typically assumes a minimum MTU of 1500 bytes. IPv6 specifications further complicate the situation because they include two design decisions that cause problems for small-MTU links. Firstly, fixed 40-byte header length adds too much overheads for IoT devices that produce small packets with little data.

(iv) Transport Layer – Transport layer in the Internet provides congestion control, guaranteed and in-order delivery of packets. These provisions are effectively used by the TCP protocol. However, TCP is not suited for IoT applications as IoT devices offer a varying traffic pattern due to their limitations. For example, due to the energy conservation requirements, IoT devices usually go into sleep mode after transmission or receiving the packet, making it difficult to maintain the communication and acknowledgement channels.

Interoperability Challenges in IoT – Heterogeneity in IoT solutions leads to Interoperability problems between IoT systems. Without interoperability, IoT deployments will be isolated to their use case and the data generated by one system can be consumed by other systems. Interoperability issues happen due to multi-vendor solutions and presence of legacy sensor network based solutions that have been deployed before the concepts of IoT was introduced.

(i)   Multi-vendor Problem – Standardization bodies only offer best practices for IoT. These recommendations only serve as guidelines for bodies that offer certification for IoT products. The certification bodies often adopt just a portion of the standards. Four well known bodies that manage certification programs today to ensure heterogeneous interoperability between IoT devices are Wi-Fi Alliance for wireless LAN technology, Bluetooth Special Interest Group (SIG) for bluetooth enabled devices, ZigBee alliance for ZigBee compatible devices and LoRa alliance for wide area IoT devices.

However, the problem lies in the fact that these certifications drive up the cost of IoT deployment and many vendors choose not to conform to the standards so that they can keep the cost down and offer value added services that can give them edge over their competitors.

(ii)   Legacy Solutions – Many vendors and solution providers have been offering closed-loop vertical solutions like building management, home automation, vehicle tracking, personnel tracking, etc. The problem with these solutions is that they were never designed to connect to the Internet so their data is contained within their domain. The challenge faced today by IoT because of these solutions is that firstly, it is too expensive to upgrade these systems to take advantage of IoT and secondly, the data models deployed in these systems are based on old database technology like relational databases, which may not conform to the concepts of non-relational big databases.

NFV has been popularized by the telecommunication companies (Telcos), who have been suffering from the interoperability problems in their core networks. In 2012, more than 20 worlds' largest telcos formed the Industry Specification Group (ISG) within the European Telecommunications Standard (ETSI) device a mechanism to virtualize their propriety core network hardware. It was aimed at addressing the operational challenges, high costs of managing closed and proprietary appliances and heterogeneity of device.

Heterogeneity in telco networks results in following issues –

(i)   Fixed Configuration – The telco hardware is configured with fixed IP locations that remain unchanged for years resulting in very rigid resource allocation.

(ii)   Manual Management – Configuration and management of telco equipment requires movement of physical staff and can be done one at a time. Hence, it is difficult to implement a common centralized policy.

(iii) *Rapid Growth of IP End Points* – Telcos offer Internet access to large number of users and this is expanding as the consumer base grows. Hence it is difficult if no central provisioning is established.

(iv) *Network End Point Mobility* – Networks are fixed and too rigid to move around so requirements for mobility takes long time to reconfigure the network for different scenarios.

(v) *Elasticity* – There is no mechanism to upgrade and downgrade a physical hardware based on demand so the networks have to be over provisioned.

(vi) *Multi-tenancy* – Usually one end-point equipment is configured for a single tenant and dynamic or multiple tenant is impossible on vendor defined hardware as vendors always wants telcos to buy more hardware.

In order to solve all these interoperability and management issues, ETSI has proposed a new architecture shown in fig. 3.4. In this new architecture, telco equipment (infrastructure layer) has been changed to dumb switches with only packet forwarding capabilities. Control layer has been separated from the hardware in order to enable centralized policy and dynamic resource allocation based on the need. Finally, application layer has been moved to the cloud to have virtually unlimited resources. This application layer could function as a firewall, DNS, load balance etc. and then communicates via software APIs to the control and infrastructure layer. Resulting in the entire core network to operate as a software that is independent of the limitations of the hardware and vendor specific protocols.
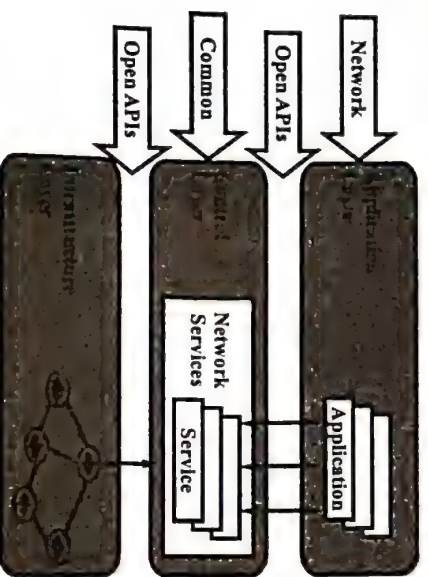

Fig. 3.4 Typical NFV Architecture

IoT today faces similar problems as Telcos did with their core network equipment and the NFV route seems like the best option for them.

Q.9. What issues might affect the development and implementation of the IoT ?

(R.G.P.V., May 2019)

Ans. The Internet of things is often lauded for its potentially revolutionary applications. Indeed, IoT devices are today being implemented in many different sectors for a vast array of purposes. However, it is still unclear how IoT will progress due to challenges associated with both technical and policy issues.

Technical Issues – Prominent technical limitations that may affect the growth and use of the IoT include a lack of new Internet addresses under the most widely used protocol, the availability of high-speed and wireless communications, and lack of consensus on technical standards.

(i) *Internet Addresses* – A potential barrier to the development of IoT is the technical limitations of the version of the Internet Protocol (IP) that is used most widely. IP is the set of rules that computers use to send and receive information via the Internet, including the unique address that each connected device or object must have to communicate.

(ii) *High-speed Internet* – Use and growth of the IoT can also be limited by the availability of access to high-speed Internet and advanced telecommunications services, commonly known as broadband, on which it depends. While many urban and suburban areas have access, that is not the case for many rural areas, for which private-sector providers may not find establishment of the required infrastructure profitable, and government programs may be limited.

(iii) *Wireless Communications* – Many observers believe that issues relating to access to the electromagnetic spectrum will need to be resolved to ensure the functionality and interoperability of IoT devices. Access to spectrum, both licensed and unlicensed, is essential for devices and objects to communicate wirelessly. IoT devices are being developed and deployed for new purposes and industries, and some argue that the current framework for spectrum allocation may not serve these new industries well.

(iv) *Standards* – Currently, there is no single universally recognized set of technical standards for the IoT, especially with respect to communications, or even a commonly accepted definition among the various organizations that have produced IoT standards or related documents. Many observers agree that a common set of standards will be essential for interoperability and scalability of devices and systems. However, others have expressed pessimism that a universal standard is feasible or even desirable, given the diversity of objects that the IoT potentially encompasses. Several different sets of de facto standards have been in development.

Several other issues affect thing the continued development and implementation of the IoT are –

(i) The lack of consensus standards for the IoT, especially with respect to connectivity.

(ii) The transition to a new Internet protocol (IPv6) that can handle the exponential increase in the number of IP addresses that the IoT will require.

(iii) Methods for updating the software used by IoT objects in response to security and other needs.

(iv) Energy management for IoT objects, especially those not connected to the electric grid.

(v) The role of the federal government, including investment, regulation of applications, access to wireless communications, and the impact of federal rules regarding "net neutrality."

6LoWPAN, IEEE 802.15.4, ZIGBEE AND ITS TYPES, RFID FEATURES, RFID WORKING PRINCIPLE AND APPLICATIONS

Q.10. Explain in brief about 6LoWPAN.

Ans. The full form of 6LoWPAN is "IPv6 over low power wireless personal area network". It operates on 2.4 GHz frequency range. It provide 250 kbit/s data rates. Fig. 3.5 shows Internet layer IPv6 sends and receive from/to adaptation layer. At adaptation layer, the data stack employs 6LoWPAN protocol before the
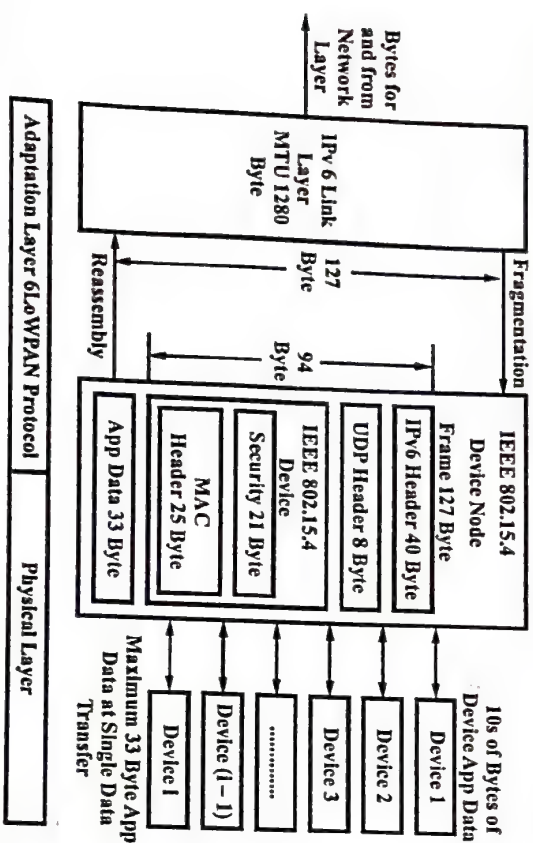
(a) Adaptation Layer 6LoWPAN Protocol   *(b) In IEEE 802.15.4 WPAN, Networked i Devices Physical Layer*

Fig. 3.5

data stack sends to IPv6 Internet layer. In case of connectivity, an IEEE 802.15.4 WPAN device contains a 6LoWPAN interface serial port. For the IEEE 802.15.4 network device, the 6LoWPAN represents an adaptation layer protocol.

The devices represent the nodes. These nodes having low speed and low power. They represent the WPAN nodes of a multiple device mesh network. Data size per instance are limited by low power devices. Data size is decreased by data compression. In addition, data size per instance is decreased by data fragmentation. Header compression, fragmentation and reassembly are the characteristics of 6LoWPAN. The first fragment header contains 27 bits which involves the datagram size and a datagram tag 16 bits, when data is fragmented before communication. Subsequent fragments include header of 8 bits. This header of 8 bits includes datagram tag, datagram size and the offset. Fragments reassembly time limit is set to 60s.

Q.11. What are the features of 6LoWPAN ?

Ans. The features of 6LoWPAN are given below –

(i) It helps mesh routing.

(ii) For reassembly of fragments, and IPv6 and UDP headers compression, neighbour discovery, the IETF recommended techniques can be specified.

Q.12. Write a brief description of IEEE standards with principles.

Ans. The Institute of Electrical and Electronics Engineers Standards Association (IEEE-SA) is an organization of (IEEE) that develops global standards in a broad range of industries, which include power and energy, biomedical and health care, information technology and robotics, telecommunication and home automation, transportation, nanotechnology, information assurance, and developed many more standards over a century, through a program that offers balance, openness, fair procedures and consensus. In the development of IEEE standards, many technical experts participate from all over the world.

The development process or the principles of IEEE standards are categorized into seven basic steps –

(i) *Securing Sponsorship* – A sponsoring organization is in charge of coordinating and supervising the standard development from inception to completion. Therefore the IEEE approved organization must sponsor a standard. All the professional societies within IEEE serve as the natural sponsor for many standards.

• (ii) *Requesting Project Authorization* – A project authorization request (PAR) is to be sent to the IEEE-SA standards board, to gain authorization for the standard.

(iii) *Assembling a Working Group* – After the approval of project authorization request, a working group of individuals affected by, or interested in, the standard is organized to develop the standards. The rules of IEEE ensure that all working group meetings are open and anyone has the right to attend and contribute to the meetings.

(iv) *Drafting the Standard* – A draft of proposed standard was prepared by the working group. This draft follows the IEEE standards style manual which sets guidelines for the clauses and format of the standards document.

(v) *Balloting* – After finalizing the draft of the standard in the working group, the draft is submitted for balloting approval. An invitation-to-ballot is send by the IEEE standards departments to any individual who has expressed an interest in the subject matter of the standard. The one who responds positively to the invitation-to-ballot becomes a member of the balloting group. The requirement of IEEE to that at least 75% of a response rate should be receive by the proposed draft. And at least 75% of the responding ballots should approve the proposed draft of the standard. In case standard is not approved, the process returns to the drafting of the standard step, in order to modify the standard document to gain approval of the balloting group.

(vi) *Review Committee* – The draft standard, along with the balloting comments are submitted to the IEEE-SA standards Board Review Committee, after getting approval of 75%. The review committee reviews the proposed draft of the standard against the IEEE-SA standard board by laws and the stipulations set forth in the IEEE-SA standards board operations manual. The review committee then makes a recommendation, whether to approve the submitted draft of the standard document.

(vii) *Final Vote* – For the submitted standard document, each member of the IEEE-SA standards board place a final vote. In some cases external members are invited to vote. To gain final approval of the standard, it takes a majority vote of the standards board. Generally, if the review committee recommends approval, the standards board will vote to approve the standard.

Q.13. Explain IEEE 802 standards for LAN.   (R.G.P.V., June 2010)

Ans. The various standards of IEEE 802 are as follows –

(i)   *802.1* – The high level interface standard addresses matters are related to network architecture, interconnection and management. Moreover, it deals with issues related to the higher OSI layers.

(ii)   *802.2* – Logical link control (LLC) and media access control (MAC) are two sublayers within 802.2 that are equivalent of the OSI data link layer.

(iii)   *802.3* – Carrier sense multiple access with collision detection (CSMA/CD) standards cover a variety of architectures that are generally based on the Ethernet as originally proposed by Metcalfe and Boggs.

(iv)   *802.4* – The token bus network standard describes how the token bus network operates.

(v)   *802.5* – The token ring network standard describes how the token ring network operates.

(vi)   *802.6* – The metropolitan area network (MAN) standard describes the operation of network covering bigger distances. Another MAN standard, the distributed queue dual bus (DQDB) by the ANSI is an adoption of the IEEE MAN standard.

(vii) *802.7* – The broadband Technical Advisory Group provides guidance to other groups that are involved in establishing broadband LAN standards.

(viii) *802.8* – The Fibre Optic Technical Advisory Group provides guidance to other groups that are involved in establishing LAN standards using fibre optic cable.

(ix) *802.9* – Integrated data and voice networks standards cover the architecture for networks that carry both voice and data like ISDNs.

(x) *802.10* – A LAN security addresses the implementation of security capability like encryption /decryption network management and data transfer.

(xi) *802.11* – The wireless LAN standard covers multiple transmission methods for wireless transmission.

(xii) *802.12* – The demand priority access method is one of the newer groups which are involved in developing specifications for 100 Mbps speed over twisted-pair wires.

(xiii) *802.14* – Cable TV based broadband communication network working group.

(xiv) *802.15* – Wireless personal area network (WPAN) working group.

(xv) *802.16* – Broadband wireless access (BBWA) working group.

(xvi) *802.17* – Resilient packet ring working group (RPRWG) for use in LAN, MAN and WAN for transfer of data packets at rates scalable to many Gbps.

Q.14. What is a IEEE 802.15.4 ?

Ans. 802.15.4 is a packet-based radio protocol. It addresses the communication needs of wireless applications that have low data rates and low power consumption requirements. It is the foundation on which ZigBee is built. The two layers specified by 802.15.4 are the physical (PHY) and MAC layers.

(i) *PHY Layers* – The PHY layer defines the physical and electrical characteristics of the network. The basic task of the PHY layer is data transmission and reception. At the physical/electrical level, this involves modulation and spreading techniques that map bits of information in such a

way as to allow them to travel through the air. Specifications for receiver sensitivity and transmit output power are in the PHY layer.

The PHY layer is also responsible for the following tasks –

(a) Enable/disable the radio transceiver

(b) Link quality indication (LQI) for received packets

(c) Energy detection (ED) within the current channel

(d) Clear channel assessment (CCA).

(ii) MAC Layer – The MAC layer defines how multiple 802.15.4 radios operating in the same area will share the airwaves. This includes coordinating transceiver access to the shared radio link and the scheduling and routing of data frames. There are network association and disassociation functions embedded in the MAC layer. These functions support the self-configuration and peer-to-peer communication features of a ZigBee network.

The MAC layer is responsible for the following tasks –

(a) Beacon generation if device is a coordinator

(b) Implementing carrier sense multiple access with collision avoidance (CSMA-CA)

(c) Handling guaranteed time slot (GTS) mechanism

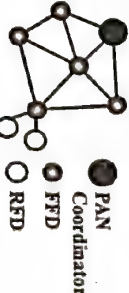(d) Data transfer services for upper layers.

Q.15. What type of devices are supported by IEEE 802.15.4 ?

Ans. There are basically two devices supported by IEEE 802.15.4. One is a full function device (FFD) and the other is a reduced function device (RFD). An FFD is equipped with full functionality. It can send, receive, route data and form clusters. It can serve as the PAN coordinator. That does not mean that it will always be a PAN coordinator. It may elect to or sometimes if it sees somebody else, some other FFD is already doing that role, then in order to save energy, it may not do PAN coordinator operation. It may just work as a full functional device and then later on if that device, which is the PAN coordinator releases its role and some other device needs to take it, then the FFD may volunteer to take that role. Next, type of device is RFD, which has a reduced functional protocol compared to the FFD protocol stack. This can only communicate to full functional devices. It cannot serve as a PAN coordinator, it serves a role as a simple sensor or switch device, and it cannot provide routing functionality. Using 802.15.4 one can form a star network topology, a P2P network topology. In addition, for a large scale network, we can form a cluster tree. Moreover, the role of coordinator is very important, it controls the 802.15.4 network. A special form of a FFD needs to be taken, and it is a typical FFD function and added on with network coordination and service features. An FFD will need to take this role of coordinator to play a bigger role in controlling the network.

Q.16. Discuss various types of network topologies used in IEEE 802.15.4.

Ans. The network topologies used in IEEE 802.15.4 are shown in table 3.1.

Table 3.1 IEEE 802.15.4 Network Topologies

S.No.	Topology	Model	Description
(i)	Star Topology		Nodes communicate via the central PAN coordinator.
(ii)	P2P Topology/ Mesh		Nodes can communicate via the PAN coordinator and via point-to-point links. It is an extension of the star topology.
(iii)	Cluster Topology		Cluster trees are similar to stars, though RFDs are only allowed to connect as the outer most branches. Diagram shows that the FFD connect the coordinator and a FFD can have connections to multiple end devices.

Q.17. What is a ZigBee and its type ?

Ans. ZigBee is the most popular industry wireless mesh networking standard for connecting sensors, instrumentation and control systems. ZigBee, a specification for communication in a wireless personal area network (WPAN), has been called the "Internet of Things". Theoretically, a ZigBee-enabled coffee maker can communicate with a ZigBee-enabled toaster. ZigBee is an open, global, packet-based protocol designed to provide an easy-to-use architecture for secure, reliable, low power wireless networks. ZigBee and IEEE 802.15.4 are low data rate wireless networking standards that can eliminate the costly and damage prone wiring in industrial control applications. Flow or process control equipment can be places anywhere and still communicate with the rest of the system. It can also be moved, since the network does not care about the physical location of a sensor, pump or valve. The ZigBee RF4CE standard enhances the IEEE 802.15.4 standard by providing a simple networking layer and standard application profiles that can be used to create interoperable multi-vendor consumer electronic solutions.

ZigBee applications include –

(i) Home and office automation

(ii) Industrial automation

(iii) Medical monitoring

(iv) Low-power sensors

(v) HVAC control

(vi) Various control and monitoring uses.

ZigBee Types –

(i) ZigBee Router (ZR) – Capable of running applications, as well as relaying information between nodes connected to it.

(ii) ZigBee End Device (ZED) – It contains just enough functionality to talk to the parent node, and it cannot relay data from other devices. This allows the node to be asleep for a significant amount of the time thereby enhancing battery life. Memory requirements and cost of ZEDs are quite low, as compared to ZR or ZC.

Q.18. Give reasons for using ZigBee.

Ans. The reasons for using ZigBee are –

(i) Reliable and self healing

(ii) Supports large number of nodes

(iii) Easy to deploy

(iv) Very long battery life

(v) Secure and low cost

(vi) Can be used globally

(vii) Open standards protocol with no or negligible licensing fees

(viii) Chipsets available from multiple sources

(ix) Remotely upgradable firmware

(x) Low maintenance.

Q.19. What are the characteristics of ZigBee ?

Ans. The characteristics of ZigBee are as follows –

(i) Global operation in the 2.4 GHz frequency band according to IEEE 802.15.4.

(ii) Regional operation in the 915 MHz (America) and 868 MHz (Europe).

(iii) Frequency agile solution operating over 16 channels in the 2.4 GHz frequency.

(iv) Incorporates power saving mechanisms for all device classes.

(v) Discovery mechanism with full application confirmation.

(vi) Pairing mechanism with full application confirmation.

(vii) Multiple star topology and inter-personal area network (PAN) communication.

(viii) Various transmission options including broadcast.

(ix) Security key generation mechanism.

(x) Utilizes the industry standard AES-128 security scheme.

(xi) Supports Alliance standards (public application profiles) or manufacturer specific profiles.

Q.20. Define the various modes of ZigBee.

Ans. ZigBee employs either of two modes, beacon or non-beacon to enable the to-and-fro data traffic. Beacon mode is used when the coordinator runs on batteries and thus offers maximum power savings, whereas the non-beacon mode is used when the coordinator is mains-powered.

(i) Beacon Mode – In this mode a device watches out for the coordinator's beacon that gets transmitted at periodically, locks on and looks for messages addressed to it. If message transmission is complete, the coordinator dictates a schedule for the next beacon so that the device 'goes to sleep'; in fact, the coordinator itself switches to sleep mode. While using the beacon mode, all the devices in a mesh network know when to communicate with each other. In this mode, necessarily, the timing circuits have to be quite accurate, or wake up sooner to be sure not to miss the beacon. This in turn means an increase in power consumption by the coordinator's receiver, entailing an optimal increase in costs.
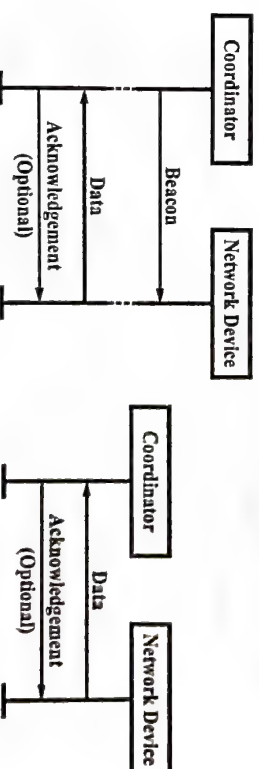
Fig. 3.6 Beacon Network Communication

(ii) Non-beacon Mode – This mode will be included in a system where devices are 'asleep' nearly always, as in smoke detectors and burglar alarms. The devices wake up and confirm their continued presence in the network at random intervals.

Fig. 3.7 Non-beacon Network Communication

Q.21. What are the network topologies supported by ZigBee.

Ans. ZigBee supports following topologies –

(i) Star Topology – The star topology consists of a coordinator placed in the centre and several end devices (nodes), as shown in the fig. 3.8 (a).

Each node is connected directly with the central coordinator. In this topology, the end devices can only communicate with the coordinator and not with other end devices. Any packet exchange between end devices can occur only through the coordinator.

(ii) Tree Topology – In this topology, the network consists of a root node which is a coordinator, several routers, and end devices, as shown in fig. 3.8 (b). All the nodes are connected in the form of the tree. The end nodes are connected directly to the coordinator and the routers as their children nodes. Both the routers and the coordinator can have children. Each end device can communicate with its parent nodes, i.e. coordinator and router. An end device cannot have children and, therefore, may not be a parent. An end device can communicate with another end device only through its parent node and there is no direct connection between end devices. Drawbacks of tree topology is that if one of the parents becomes disabled, the children of the disable parent cannot communicate with other devices in the network.

(iii) Mesh Topology – Mesh topology is also known as peer to peer topology. A ZigBee mesh network consists of three types of nodes – a coordinator, several routers, and one or more end devices. The coordinator can send packets to any node in the network. If the node is not in range, the message will be sent to a neighbouring node which will then forward it onward to the destination. The mesh network can cover a larger range while using only a fraction of power. The ZigBee mesh network is capable of growing or shrinking depending on one's needs just be adding or removing nodes. A mesh topology is self-healing, i.e. during transmission, if any of the path fails, the node will find an alternate path to the destination. Adding or removing a device is easy. Any source device can communicate with any destination device in the network. Compared with star topology, mesh topology requires greater overhead. Mesh routing uses a more complex routing protocol than a star topology.

(a) Star Topology

(b) Tree Topology

(c) Mesh Topology
Fig. 3.8
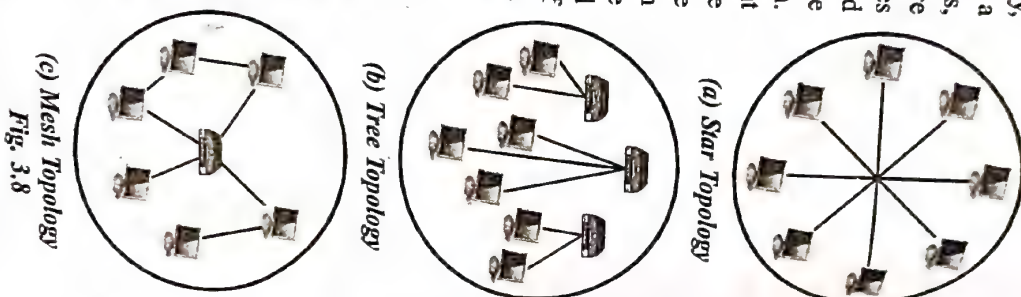
Q.22. Describe ZigBee architecture in detail.

Ans. The ZigBee architecture is shown in fig. 3.9.
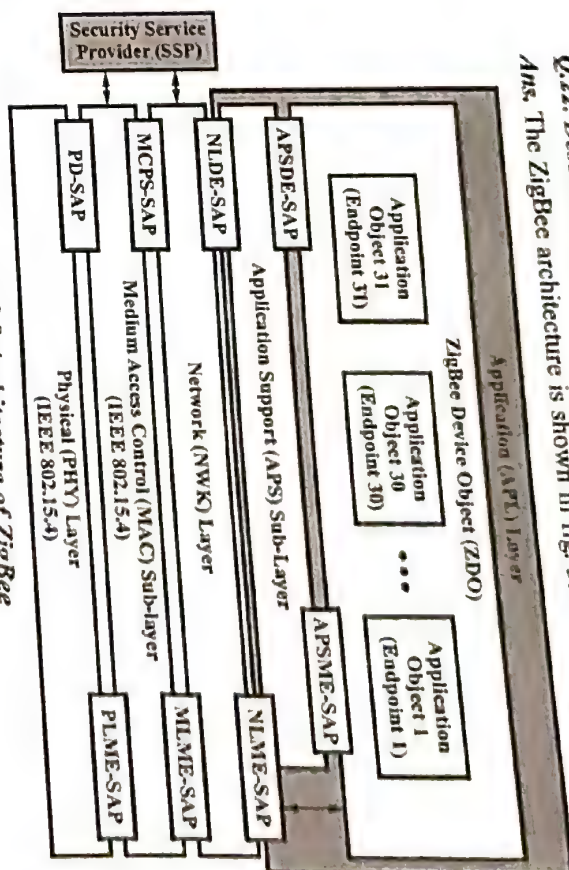
Fig. 3.9 Architecture of ZigBee

Various components of ZigBee are discussed below –

(i) Physical (PHY) Layer – Refer to Q.14 (i).

PHY layer supports three frequency bands, 2.45 GHz band which using 16 channels, 915 MHz band which using 10 channels and 868 MHz band using 1 channel. All three using Direct Spread Spectrum Sequencing (DSSS) access mode.

PHY packet fields are –

(a) Preamble (32 bits) – synchronization
(b) Start of packet delimiter (8 bits)
(c) PHY header (8 bits) – PSDU length
(d) PSDU (0 to 1016 bits) – Data field.

Preamble	Start of Packet Delimiter	PHY Header	PHY Service Data Unit (PSDU)

Fig. 3.10 Packet Structure

(ii) MAC Layer – Refer to Q.14 (ii).

This layer provides interface between physical layer and network layer. This provides two services – MAC data services and MAC management service interfacing to the MAC sub Layer Management Entity (MLME) Service Access Point called (MLMESAP). The MAC data service enables the transmission

and reception of MAC Protocol Data Units (MPDUs) across the PHY data service. Basically there are two types of topology – star and peer to peer. Peer to peer topology can take different shapes depending on its restrictions. Peer to peer is known as mesh, if there is no restriction. Another form is tree topology. Interoperability is one of the advantages of ZigBee protocol stack. ZigBee has wide range of applications, so different manufacturer provides ZigBee devices. ZigBee devices can interact with each other regardless of manufacturer (even if the message is encrypted).

(iii) *Network (NWK) Layer* – Network layer interfaces between application layer and MAC Layer. This layer is responsible for network formation and routing. Routing is the process of selection of path to relay the messages to the destination node. This forms the network involving joining and leaving of nodes, maintaining routing tables (coordinator/router), actual routing and address allocation. ZigBee coordinator or router will perform the route discovery. This layer provides network wide security and allows low power devices to maximize their battery life. From the basic topologies, there are three network topologies as considered in IEEE802.15.4 which are star network, tree network and mesh network.

(iv) *Application (APL) Layer* – This layer is the highest protocol layer and it hosts the application objects. ZigBee specification separates the APL layer into three different sublayers – the application support sub layer, the ZigBee device objects, and application framework having manufacturer defined application objects.

(a) **The Application Objects (APO)** – These objects control and manages the protocol layers in ZigBee device. It is a piece of software which controls the hardware. Each application objects assigned unique end point number that other APO's can use an extension to the network device address to interact with it. There can be up to 31 application objects in a single ZigBee device. A ZigBee application must conform to an existing application profile which is accepted ZigBee Alliance. An application profile defines message formats and protocols for interactions between application objects. The application profile framework allows different vendors to independently build and sell ZigBee devices that can interoperate with each other in a given application profile.

(b) **ZigBee Device Object** – The key definition of ZigBee is the ZigBee device object, which addresses three main operations – service discovery, security and binding. The role of discovery is to find nodes and ask about MAC address of coordinator/router by using uncast messages. The discovery is also facilitating the procedure for locating some services through their profile identifiers. So profile plays an important role. The security services through keys for data encrypt... . The network manager is i... lemented in the

coordinator and its role is to select an existing PAN to interconnect. It also supports the creation of new PANs. The role of binding manager is to binding nodes to resources and applications also binding devices to channels.

(c) **Application Support Sub Layer** – The application support (APS) sub layer provides an interface between the NWK and the APL layers through a general set of services provided by APS data. The APS sub layer processes outgoing/incoming frames in order to securely transmit/receive the frames and establish the cryptographic keys. The upper layers issue primitives to APS sub layer to use its services. APS layer security includes the following services – Establish Key, Transport Key, Update Device, Remove Device, Request Key, Switch Key, Entity Authentication, and Permissions Configuration Table.

(vi) *Security Service Provider* – ZigBee provides security mechanism for network layer and application support layers, each of which is responsible for securing their frames. Security services include methods for key establishment, key transport, frame protection and device management.

Q.23. Enlist security services provided in ZigBee.

Ans. Security and data integrity are key benefits of the ZigBee technology. ZigBee leverages the security model of the IEEE 802.15.4 MAC sub layer which specifies four security services –

(i) Access control by maintaining a list of trusted devices within the network.

(ii) Data encryption, which uses symmetric key 128-bit advanced encryption standard.

(iii) Frame integrity to protect data from being modified by parties without cryptographic keys.

(iv) Sequential freshness to reject data frames that have been replayed – the network controller compares the freshness value with the last known value from the device and rejects it if the freshness value has not been updated to a new value.

The actual security implementation is specified by the implementer using a standardized toolbox of ZigBee security software.

Q.24. Describe traffic types handled in ZigBee/IEEE 802.15.4.

Ans. ZigBee/IEEE 802.15.4 addresses three typical traffic types. IEEE 802.15.4 MAC can accommodate all three types as follows –

(i) *Data is Periodic* – The application dictates the rate, and the sensor activates, checks for data and deactivates.

(ii) *Data is Intermittent* – The application, or other stimulus, determines the rate, as the case of say smoke detectors. The device needs to connect to the network only when communication is necessitated. This type enables optimum saving on energy.

(iii) *Data is Repetitive* – When data is repetitive and the rate is fixed a priori, then depending on allotted time slots, called GTS (guaranteed time slots), devices operate for fixed durations.

Q.25. Write short note on media access control. (R.G.P.V., May 2018)

Or

Explain media access control. **(R.G.P.V., May 2019)**

Ans. Each node which is connected to the network, has an MAC address, A device node obtains data stack with its MAC address. Node represents an IoT device (sensor or computer), the data link layer of which communicates to the Internet. MAC address is 48 bit. Every network card has a unique MAC address using a communicating node in the destination and source node address. Ethernet frame uses before the data stack destination node MAC address and source node MAC address. In the firmware of the network card or chip. MAC address of each node can be defined. Every node to be addressed by the MAC address is enabled by ARP. The network data can be obtained with the node from a router having an IP address. Every node is enabled by using RARP for transmitting on the network the data to a router having an IP address. Lookup table is used by address resolution protocol. Using that table, the network 32-bit address gives MAC address of the single node. This lookup table is also used by RARP. The table is used to store the IP address in one column. MAC address is stored in each row of another column. Number of rows is equal to the number of nodes connected to the Internet. Using a table, a single node MAC address gives the network a 32-bit address for the Internet. For the address translation of the data stack obtained at the Internet layer to the node MAC address, ARP cache works like building up of the table. They serves a cache called ARP cache, whenever the first time a node transmits the sender MAC and IP addresses or the network transmits the receiver IP and MAC addresses. For the next communication between the node and the network, the cached addresses create a lookup table. At the new IP address, the process enables the node to send or receive the IP packets from the Internet with the help of address resolution.

Q.26. Write short note on TCP.

(R.G.P.V., June 2011)

Ans. Transmission control protocol (TCP) provides a connection-oriented user-to-user byte stream service. This means it provides a logical connection between two sites and is capable of transmitting a sequence of bytes between them. It divides a byte stream into a sequence of segments and sends them to the destination via a variation on a sliding window flow control protocol. It provides the initial handshaking by establishing, maintaining, and releasing connections. It handles requests to deliver information to a destination reliably, an important consideration since the lower layer does not guarantee about

delivery of packets. TCP receives data or requests from its user, stores it in a TCP segment format and gives it to the IP, which is shown in fig. 3.11. It plays no role in the subsequent routing and transfer of information.
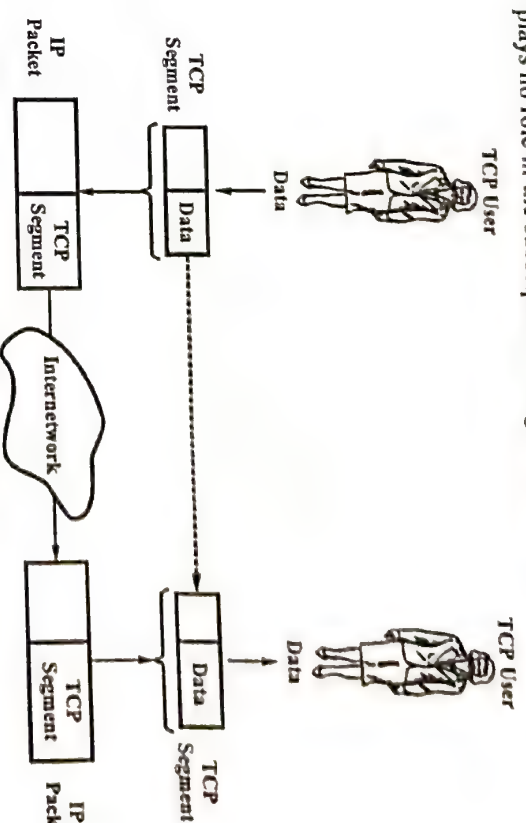
Fig. 3.11 TCP as a User-to-user Service

Each byte on a TCP connection its own 32-bit sequence number. In case of a host blasting away at full speed on a 10 Mbps LAN, theoretically the sequence numbers could wrap around in an hour, but practically it takes much longer. Sequence numbers are used both for acknowledgement and for the window mechanism, in which separate 32-bit header fields are used.

Q.27. Write short note on UDP.

(R.G.P.V., June 2004)

Ans. UDP (user datagram protocol) is another transport layer protocol that is a part of TCP/IP suite. It is an unreliable, connectionless protocol that does not guarantee delivery and duplicate protection. UDP is basically just IP with a short header added. It provides non-sequential transport functionality when reliability and security are less important than size and speed.

UDP provides a way for applications to send encapsulated raw IP datagrams without establishing any connection. UDP is more appropriate than TCP in some contexts; like it is more robust at lower layers. UDP has very little to do. Mainly, it adds a port addressing capability to IP. Since its header is shorter, it involves less overheads.

UDP is used by the applications that do not want TCP's sequencing and flow control and wish to provide their own. It is widely used for one-shot, client-server type request-reply queries and applications in which prompt delivery is more important than accurate delivery, such as transmitting speech

or video. Thus, there is a place at the transport layer for both connection-oriented and connectionless type of service. UDP is described in RFC 768.

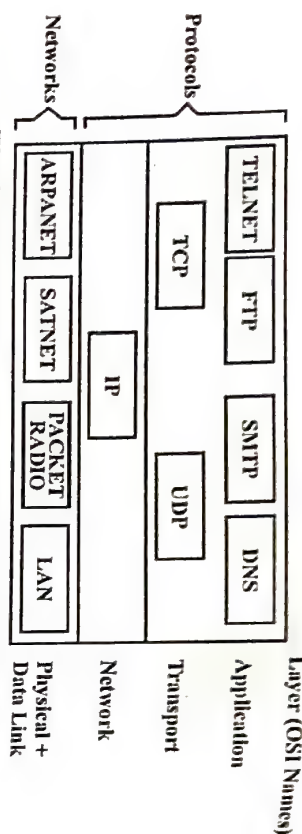The relation of IP, TCP and UDP is shown in fig. 3.12.

Layer (OSI Names)	Protocols				
Application	TELNET	FTP	SMTP	DNS	
Transport	TCP		UDP		
Network	IP				
Physical + Data Link	ARPANET	SATNET	PACKET RADIO	LAN	

Networks

Fig. 3.12 Protocols and Networks in TCP/IP Model

Q.28. Write short note on RFID.

(R.G.P.V., May 2019)

Ans. The radio frequency identification (RFID) is a unique identity of object or person wirelessly using radio waves in the form of numbers. RFID technology plays an important role in IoT for solving identification issues. RFID system is composed of one or more reader and several RFID tags. Tags uses radio-frequency electromagnetic fields to transfer data attached to an object. The tags contain electronically stored information. Passive tags collect energy from a nearby RFID reader's interrogating radio waves. The RFID device serves the same purpose as a bar code or a magnetic strip on the back of a credit card or ATM card; it provides a unique identifier for that object. And, just as a bar code or magnetic strip must be scanned to get the information, the RFID device must be scanned to retrieve the identifying information.

Q.29. Discuss the use of RFID tags in IoT systems.

Ans. Using a RFID (radio frequency identification) tag an object can be identified at different locations and times. A product, parcel, postal article, person, bird, animal, vehicle or object can be tagged. A RFID tag can be identified by the reader circuit of an ID using UART or NFC protocol, from a distance of 20 m. An active device which has an in-built source of power such as an NFC enabled mobile, generates an RF field which induces the current in RFID tag and generates enough power for radio frequency identification. Using that power, the device transmits the contents of RFID tag.

In case of a passive device, the power is obtained from the electrical current in its antenna by incoming RF signals from a reader or hotspot and the tag information is then transmitted back.

A hotspot has a wireless or Wi-Fi transceiver for Internet connectivity, which receives signals from a number of RFID tags in a system and transmits the data to a web server over the Internet. RFIDs thus form a network, which

is connected to an IoT server through Internet. An IoT server consists a RFID identity manager, a device manager, database server for data storage and fetching, a data analyser and a whole lot of services.

IoT Applications using RFID – RFID founds application for tracking business processes like for payment, leasing, insurance, quality management, inventory control of goods, supply chain systems and devices like RFID based temperature or any other parameter sensor. Some advanced applications of RFID include factory designing, taking protective measures for anti-counterfeiting.

Q.30. How does radio frequency identification (RFID) work ? Explain.

Ans. RFID systems uses three components in two combinations – a transceiver (transmitter/receiver) and antenna are usually combined as an RFID reader. A transponder (transmitter/responder) and antenna are combined to make an RFID tag. An RFID tag is read when the reader emits a radio signal that activates the transponder, which sends data back to the transceiver.

Three basic components used by a RFID system are –

   (i) An antenna or coil.

   (ii) A transceiver (with decoder).

   (iii) A transponder (RF tag) electronically programmed with unique information.

There are two types of transponders, which correlate to the two major types of RFID tags.

Passive transponders and RFID tags have no energy source of their own, relying on the energy given off by the reader for the power to respond. Cheaper, passive RFID tags are the most likely to be used for consumer goods.

An active transponder or tag has an internal power source, which it uses to generate a signal in response to a reader. Active transponders are more expensive than passive ones. They can communicate over miles like ordinary radio communications. They are commonly used in navigation systems for commercial and private aircraft.

There are many uses of this technology around us today, although they are often invisible to users. You may find that you are already carrying and using a RFID tag, or even several. At its most basic level, RFID is a wireless link to uniquely identify objects or people. It is sometimes called dedicated short range communication (DSRC). RFID systems include electronic devices called transponders or tags, and reader electronics to communicate with the tags. These systems communicate via radio signals that carry data either unidirectional or bidirectional. As shown in fig. 3.13, when a transponder enters a read zone, its data is captured by the reader and can then be transferred through standard interfaces to a host computer, printer, or programmable logic controller for storage or action.
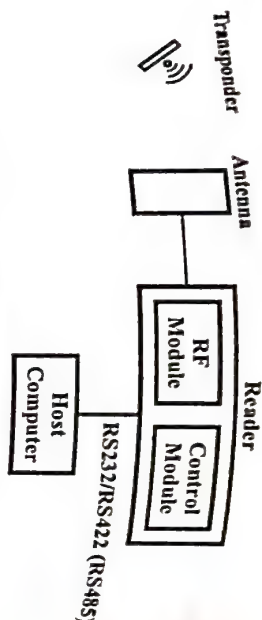

Transponder   Antenna   **Reader**

Fig. 3.13 Working of RFID

The antenna emits radio signals to activate the tag and to read and write data to it. The reader emits radio waves in ranges of anywhere from one inch to 100 feet or more, depending upon its power output and the radio frequency used. When an RFID tag passes through the electromagnetic zone, it detects the reader's activation signal. The reader decodes the data encoded in the tag's integrated circuit (silicon chip) and the data is passed to the host computer for processing.

The purpose of an RFID system is to enable data to be transmitted by a portable device, called a tag, which is read by an RFID reader and processed according to the needs of a particular application. The data transmitted by the tag may provide identification or location information, or specifics about the product tagged, such as price, color, date of purchase, etc. RFID technology has been used by thousands of companies for a decade or more. RFID quickly gained attention because of its ability to track moving objects. As the technology is refined, more pervasive and invasive uses of RFID tags are in the works.

A typical RFID tag consists of a microchip attached to a radio antenna mounted on a substrate. The chip can store as much as 2 kilobytes of data.

To retrieve the data stored on an RFID tag, a reader is needed. A typical reader is a device that has one or more antennas that emit radio waves and receive signals back from the tag. The reader then passes the information in digital form to a computer system.

Once a link is established with a unique ID on an item, then automation of an assortment of processes ensures.

Q.31. What are the benefits of RFID ?

Ans. The benefits of RFID are as follows –

(i) Tag detection not requiring human intervention reduces employment costs and eliminates human errors from data collection.

(ii) As no line-of-sight is required, tag placement is less constrained.

(iii) RFID tags have a longer read range than other techniques such as barcodes.

(iv) Tags can have read/write memory capability, while barcode do not

(v) An RFID tag can store large amounts of data additionally to a unique identifier.

(vi) Unique item identification is easier to implement with RFID than with barcodes.

(vii) It has ability to identify items individually rather than generically.

(viii) Tags are less sensitive to adverse conditions (dust, chemicals, physical damage etc.).

(ix) Many tags can be read simultaneously.

(x) RFID tags can be combined with sensors.

(xi) Automatic reading at several places reduces time lags and inaccuracies in an inventory.

(xii) Tags can locally store additional information, such as distributed data storage may increase fault tolerance of the entire system.

(xiii) Reduces inventory control and provisioning costs.

(xiv) Reduces warranty claim processing costs.

Q.32. Explain the principle of RFID.

Ans. An RFID tag is an electronic circuit which transmits its ID using RF signals. When tag comes in contact with the RF signals, it transmits back a short string of data to the reader. The information so received by the RFID reader is communicated to a web server or cloud server over the Internet. Some additional information may also be send by the reader depending upon the application.

An RFID tag is simpler to process as compared to a bar code or QR code. Further, it uses short range RF transceivers, which makes them invisible.

Q.33. Explain in brief about components of an RFID system.

Ans. Various components of an RFID system used for IoT applications (as shown in fig. 3.14) are –

(i)   Tag   (ii) Transceiver
(iii) Reader   (iv) Data processing subsystem
(v)   Middleware   (vi) Applications and services.

(i) **Tag** – A tiny chip, which can receive RF signals, functions as a tag. A chip used on objects can be of following three types –

(a) Active chip
(b) Passive chip
(c) Battery powered passive chip.

(ii) **Transceiver** – A transceiver is not a separate component, it is in-built at the chip. Depending upon the chip used it can communicate with the reader in a range of 10 cm to 200 m. The chip can communicate with the

reader by UART communication either using RF links or NFC. frequency range for operation of these transceivers are 120-150 kHz, Standard MHz, 433 MHz, higher when using UHF and microwave frequencies.

Middleware

Computing Device → Reader Transceiver → RFID Transceiver

Internet → Hotspot → Transceiver Reader → RFID Transceiver

Datacenter/ Cloud/Web Server — Data Store

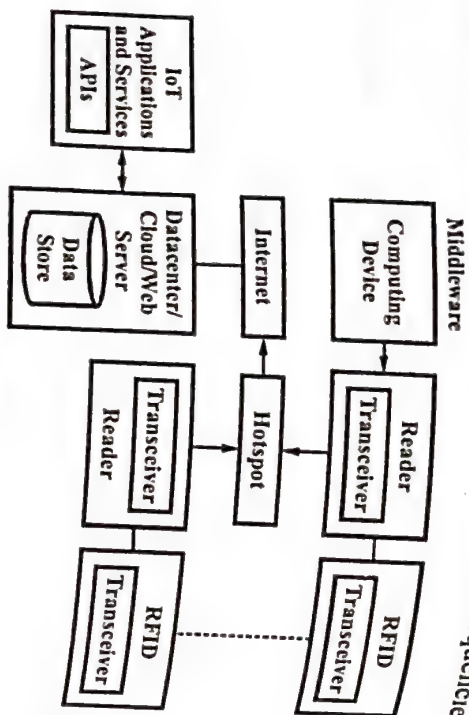IoT Applications and Services — APIs

Fig. 3.14

(iii) Reader – A reader which is used to read an ID tag, has a transceiver within it. If device is using UART protocol, the data received will consists of 1 start byte, a 10 byte ID and 1 end byte. The signals received from the RFID tag are sent over the Internet.

(iv) Data Processing Subsystem – A data processing subsystem consists of a computing device and a middleware. It provides Internet connectivity, directly through a gateway. The subsystem acts a backend system.

(v) Middleware – These are the software components used at the reader, read manager, data store for the transactional data store and APIs of the applications.

Q.34. What are the issues with the use of RFID technology?
Ans. RFID technology, which appears to be very simple has following issues –

(i) Design Issue – A standard global framework is required to design a unique ID.

(ii) Security Issue – A tag being read only, can interact with any reader and thus can be tracked without authority. To ensure privacy and security, data processing at the tag and reader, need to be supplemented with access encryption and authentication algorithms. RFID systems are also vulnerable to virus attacks.

(iii) Cost Issue – Cost of RFID tag and reader increases when provided with data processing and security enhancing technique.

condition, thus needs protection.

(iv) Protection Issue – The tag can be damaged in adverse weather condition, thus needs protection.

(v) Active Life Issue – Active RFID devices, which consists battery have a life of only 2 to 4 years.

(vi) Recycling Issue – Recycling of tags can harm the environment.

Q.35. How does an RFID tag securely communicate and read by a reader? How is an RFID tag secured? (R.G.P.V., May 2018)
Ans. An radio frequency identification device tag may have a memory and microprocessor. Then it computes one way hash function known as meta ID on tag. If the radio frequency identification device reader uses the meta-ID, then only the tag communication unlocks and the tag becomes readable. After the reader finishes reading, the tag gets locked. This disables reading by unknown entities to the system. The tag can self-destruct if under attack.

Q.36. Enlist various technological challenges with the implementation of RFID technology.
Ans. Various technological challenges with the RFID are given below –
(i) RFID hotspot needs wireless installation, the frequency of which may interfere with the frequencies of existing wireless systems in the organization.
(ii) Data processing subsystem which consists reader and tag protocols, middleware architecture and EPC standards needs to be implemented effectively.
(iii) Tags and RFID technology are costly.
(iv) Security of data.
(v) Design robustness.

Q.37. What are the security issues associated with RFID?
Ans. The security issues associated with RFID technology are given below –
(i) Maintain overall data integrity and identify foreign attacks.
(ii) Preventing cloning of the tag by an unauthorised entity.
(iii) Unauthorised tag manipulation by an external reader can make the tag useless.
(iv) An unauthorised external reader can disable the tag.
(v) An external reader may pretend to be a part of the system, thus can affect the privacy of data.
(vi) An external object may pretend to be a system tag or reader.

Q.38. How is the IPv6 used for the RFID objects using ORCHID?
Ans. The data captured by a RFID reader after filtering, aggregation, and routing get stored at an IP (Internet Protocol) address in XML format. This data can be accessed using HTTP and SOAP protocols.

For communicating (or routing) the data captured by the reader over the Internet, an IPv6 protocol is used. IPv6 is a 128-bit IP address, which needs to be mapped with the 96-bit EPC. The EPC contains header, manufacturer, product and serial number bits.

In order to secure this IPv6 communication, Cryptographically Generated Addresses (CGAs) are used. In this method a cryptographic one-way hash function is generated by a host suffix/interface identifier. The input to this function is a public key as binary input. CGA being unique is a very secure method for IPv6 communication. The CGAs however cannot be used at the IP layer for RFID IoT systems.

In an alternate method, a new class of identifiers based on CGAs are used for IP communication. These identifiers are known as Overlay Routable Cryptographic Hash Identifiers (ORCHID). The ORCHID format is compatible with the IPv6-like address format. However, IPv6 has a 64-bit network prefix locator to identify, the network locations, while ORCHID format does not have a locator. The ORCHID identifier is only used by the APIs and applications.

NFC (NEAR FIELD COMMUNICATION), BLUETOOTH WIRELESS SENSOR NETWORKS AND ITS APPLICATIONS

Q.39. Explain in detail about the NFC (near field communication).

Ans. Near field communication is some how little bit similar to RFID, it combines a RFID reader in a mobile phone, which makes it better, reliable and efficient for the users. Near field communication is a short-range wireless technology with the frequency of 13.56 MHz, typically work for very small distance up to 4 cm. Allows intuitive initialization of wireless networks and NFC is complementary to bluetooth and 802.11 with their long distance capabilities at a distance circa up to 10 cm. It is first developed by Philips and Sony companies. Data exchange rate was approximately 424 kbps. Power consumption during data reading in NFC is under 15 ma. There are two modes in NFC technology, viz. Active and Passive.

(i) Active Mode – In this mode both the devices are active and communicate with each other by sending the signals.

(ii) Passive Mode – In this mode one of the device sends the signal rather other just receiving it.

NFC does not need pairing, it cannot work from a long distance and in this way this technology is secure and used for mobile payments.

Application – NFC works in a very short range so the devices must be kept nearby. It has several applications, the most important one is Payment

App. Today, we have several applications (apps) by which one can pay without using a card, in this scenario the device works as a virtual card and the transaction takes place. One can exchange their business card with the help of their devices. They just touch their devices and their business cards will be exchanged. If a information is required than use the device with the smart poster and get all the information with a single touch. It can also work while travelling, a person can book a travel ticket or a room in a hotel. While booking keys are given to the person, when person touch the device on the appropriate devices, the work is done and the person will move in.

Issues – These devices will work on a very small range, so this is one of the major issues. Two devices of two different manufacturers can create a monopoly some compatibility issue in their communication. Due to this reason a monopoly may exist in market.

Q.40. Explain architecture of Bluetooth networks. Discuss various applications of Bluetooth network.

Ans. The L.M. Ericsson Company became interested in connecting its mobile phones to other devices (e.g., PDAs) without cables in 1994. Four other companies (IBM, Intel, Nokia and Toshiba), made a SIG (special interest group, i.e., consortium) to develop a wireless standard for interconnecting computing and communication devices and accessories with the aid of short-range, low-power, inexpensive wireless radios. The project was named *Bluetooth*, after Harald Blaatand (Bluetooth) II (940-981), a Viking king who unified started Denmark and Norway, also without cables.

Although the original idea was just to overcome the cables between devices, it soon started to expand in scope and encroach on the area of wireless LANs. While this move makes the standard more useful, it also creates some competition for mindshare with 802.11. The two systems also interfere with each other electrically, to make matters worse. It is also noticed that Hewlett-Packard introduced an infrared network for connecting computer peripherals without wires some years ago, but it never really caught on in a bit way.

The main aim of Bluetooth is to develop a single-chip, low-cost, radio-based wireless network technology.

At the same time the Bluetooth development started, a study group within IEEE 802.11 wireless personal area network (WPAN) discussed under the following five criteria –

(i) Distinct Identity – Originally the study group did not want to establish a second 802.11 standard.

(ii) Compatibility – Compatibility with IEEE 802 standard.

(iii) Market Potential – How many applications, devices, vendors, customers are available for a certain technology ?

(iv) Technical Feasibility – Prototypes are required for further discussion, so the study group would not rely on paper work.

(v) Economic Feasibility – Everything developed within this group should be cheaper compared to other solutions and permit for high-volume production.

Bluetooth meets these criteria so the WPAN group cooperated with the Bluetooth consortium.

To understand the networking of Bluetooth devices a quick introduction to its key features is necessary.

The *Piconet* is the basic unit of Bluetooth system. It consists of a master node and upto seven active slave nodes within a distance of 10 meters. Multiple piconets can be present in the same (large) room and can even be connected through a bridge node as shown in fig. 3.15. An interconnected collection of piconets is known as a *scatternet.*
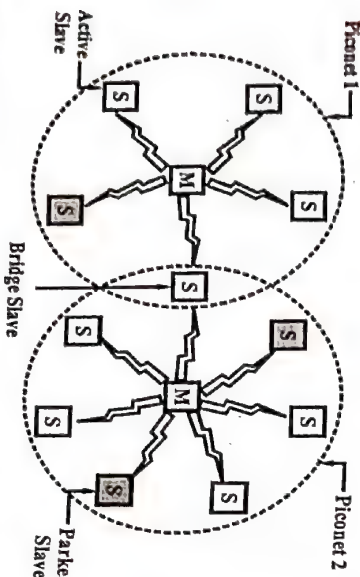

Fig. 3.15 Two Piconets can be Connected to Form a Scatternet

There can be upto 255 parked nodes in the net for the seven active slave nodes in a piconet. These are devices that the master has switched to a low-power state to reduce the drain on their batteries. In parked state, a device cannot do anything except respond to an activation or beacon signal from the master. Also, two intermediate power states are hold and sniff.

The reason behind the master/slave design is that the designers intended to make easy the implementation of complete Bluetooth chips for under $5. The result of this decision is that the slaves are fairly dumb, basically just doing whatever the master tells them to do. Therefore, a piconet is a centralized TDM system, with the master controlling the clock and determining which device gets to communicate in which time slot. All communication is between the master and a slave; direct slave-slave communication is not possible.

Fig. 3.16 shows an overview of the formation of a piconet. As all active devices have to use the same hopping sequence they must be synchronized.
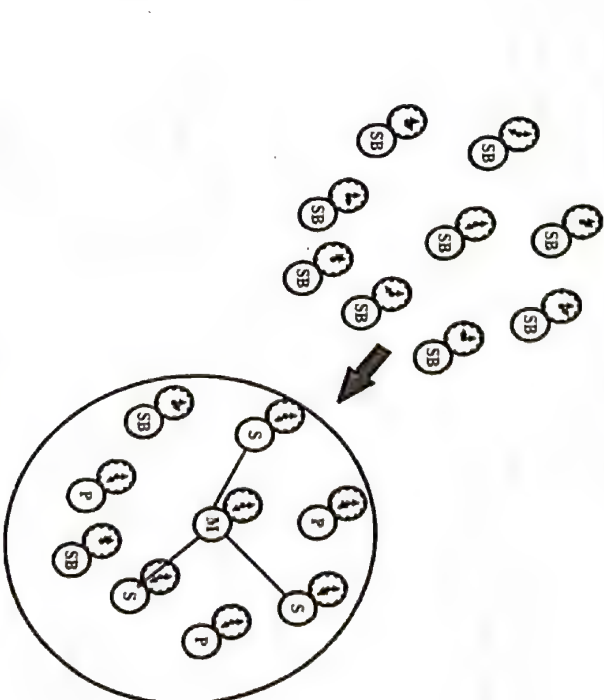
The first step involves a master sending its clock and device ID. All Bluetooth devices have the same networking capabilities, i.e., they can be master or slave. There is no distinction between terminals and base stations, any two or more devices can form a piconet. The unit establishing the piconet automatically becomes the master, all other devices will be slaves. The hopping pattern is determined by the device ID, a 48-bit worldwide unique identifier. After adjusting the internal clock according to the master a device may participate in the piconet. All active devices are assigned a 3-bit active member address (AMA). All parked devices use an 8-bit parked member address (PMA). Devices in stand-by do not need an address.

Fig. 3.16 Formation of Piconet

Any user within one piconet shares the same 1 MHz channel. As more users join the piconet, the throughput per user drops quickly (a single piconet offers less than 1 Mbit/s gross data rate). This led to the idea of forming groups of piconets called scatternet. Only those units that really must exchange data share the same piconet, so that many piconets with overlapping coverage can exist simultaneously.

Bluetooth applies FH-CDMA for separation of piconets. In an average sense, all piconets can share the total of 80 MHz bandwidth available. Adding more piconets leads to a graceful performance degradation of a single piconet because more and more collisions may occur. A collision occurs if two or more piconets use the same carrier frequency at the same time. This will probably happen as the hopping sequences are not co-ordinated.

If a device wants to participate in more than one piconet, it has to synchronize to the hopping sequence of the piconet it wants to take part in. If a device acts as slave in one piconet, it simply starts to synchronize with the hopping sequence of the piconet it wants to join. After synchronization, it acts as a slave in this piconet and no longer participates in its former piconet. To enable synchronization, a slave has to know the identity of the master that determines the hopping sequence of a piconet. Before leaving one piconet, a slave informs the current master that it will be unavailable for a certain amount of time. The remaining devices in the piconet continue to communicate as usual.

Applications – 802.11 does not specify whether users should use their notebook computers for reading e-mail, surfing the web, or something else. In contrast, the Bluetooth V1.1 specification names 13 specific applications to be supported and provides different protocol stacks for each one. Unfortunately, this approach leads to a very large amount of complexity, which we will omit here. The 13 applications, which are called *profiles*, are listed in fig. 3.17.

Name	Description
Generic access	Procedures for link management
Service discovery	Protocol for discovering offered services
Serial port	Replacement for a serial port cable
Generic object exchange	Defines client-server relationship for object movement
LAN access	Protocol between a notebook computer and a fixed LAN
Dial-up networking	Allows a notebook computer to call via a mobile phone
Fax	Allows a mobile fax machine to talk to a mobile phone
Cordless telephony	Connects a handset and its local base station
Intercom	Digital walkie-talkie
Headset	Allows hands-free voice communication
Object push	Provides a way to exchange simple objects
File transfer	Provides a more general file transfer facility
Synchronization	Permits a PDA to synchronize with another computer

Fig. 3.17 The Bluetooth Profiles

The first one generic access profile is not really an application, but rather the basis upon which the real applications are built. Its main job is to provide a way to establish and maintain secure links between the master and slaves. Also relatively generic is the service discovery profile, which is used by devices to discover what services other devices have to offer. These two profiles are necessary for all Bluetooth devices. The remaining one are optional.

The serial port profile is a transport protocol that most of the remaining profile use. It emulates a serial live and useful for legacy applications. The generic object exchange profile defines a client-server relationship for moving data around. The next three profiles is for networking. The LAN access profile allows a Bluetooth device to connect to a fixed network. This profile is

---

a direct competitor to 802.11. The dial-up network protocol is used to allow a notebook computer to connect to a mobile phone containing a built-in modem without wires.

The fax profile is similar to dial-up networking except that it allows wireless fax machines to send and receive faxes using mobile phones without a wire between the two.

The next three profiles cordless telephony, intercom, headset profiles are for telephony. The remaining three profiles are for actually exchanging objects between two wireless devices. These could be business cards, pictures, or data files. The synchronization profile, in particular, is intended for loading data into a PDA or notebook computer when it leaves home and collecting data from it when it returns.

Q.41. Explain different user scenarios for a Bluetooth network.

Ans. Many different user scenarios are there for wireless piconets or WPANs –

(i) Connection of Peripheral Devices – At present, most devices are connected to a desktop computer via wires. This type of connection has several disadvantages – each device has its own type of cable, different plugs are needed, wires block office space. In a wireless network, no wires are needed for data transmission. However, batteries now have to replace the power supply, as the wires not only transfer data but also supply the peripheral devices with power.

(ii) Support of Ad-hoc Networking – Imagine several people coming together, discussing issues, exchanging data. For instance, students might join a lecture, with the teacher distributing data to their personal digital assistants (PDAs). Wireless networks can support this type of interaction; small devices might not have WLAN adapters following the IEEE 802.11 standard, but cheaper Bluetooth chips built in.

(iii) Bridging of Networks – Using wireless piconets, a mobile phone can be connected to a PDA or laptop in a simple way. Mobile phones will not

Fig. 3.18 Example Configurations with a Bluetooth-based Piconet

have full WLAN adapters built in, but could have a Bluetooth chip. The mobile phone can then act as a bridge between the local piconet and, e.g., the global GSM network (see fig. 3.18). For instance, on arrival at an airport, a person's mobile phone could receive e-mail via GSM and forward it to the laptop which is still in a suitcase. Via a piconet, a file server could update local information stored on a laptop or PDA while the person is walking into the office.

Q.42. Discuss the protocol architecture of Bluetooth.

Ans. The Bluetooth standard has many protocols grouped loosely into layers. The basic Bluetooth architecture is shown in fig. 3.19. Starting as a simple idea, it now covers over 2,000 pages dealing with not only the Bluetooth protocols stack can be divided into a core specification (Bluetooth, 2001 a), which describes the protocols from physical layer to data link control together with management function, and profile specifications (Bluetooth, 2001 b).
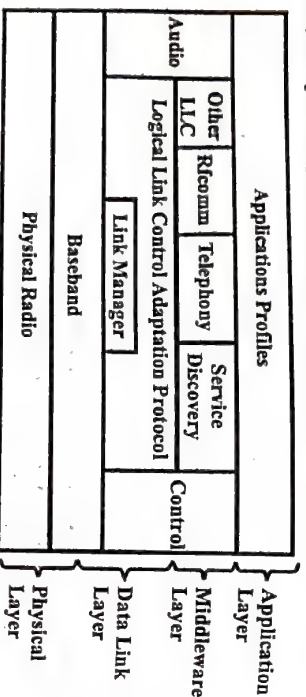
Applications Profiles					Application Layer
Audio	Other	Rfcomm	Telephony	Service Discovery	Middleware Layer
	LLC				
	Logical Link Control Adaptation Protocol			Control	
	Link Manager				Data Link Layer
	Baseband				
	Physical Radio				Physical Layer

Fig. 3.19 The 802.15 Version of the Bluetooth Protocol Architecture

The bottom layer is the physical radio layer, which corresponds fairly well to the physical layer in the OSI and 802 models. This layer deals with radio transmission and modulation.

Next the baseband layer is somewhat analogous to the MAC sublayer but also includes elements of the physical layer. It deals with how the master controls time slots and how these slots are grouped into frames.

Next comes a layer with a group of somewhat related protocols. The link manager set up the logical channels between devices, including security functions and parameter negotiation. The logic link control adaption protocol shields the upper layer from the details of transmission. Audio and control layer deals with audio and control respectively. The applications can get at them directly, without having to go through the L2CAP protocol.

The next layer up is the middleware layer. Which contains a mix of different protocols.

The top layer is where the applications and profiles are located. They make use of the protocols in lower layers to get their work done. Each application has

its own dedicated subset of the protocols. Specific devices, such as a headset, usually contain only those protocols needed by that application and no others.

The three lowest layers of the Bluetooth protocol stack are discussed below. These layers roughly correspond to the physical and MAC sublayers.

(i)   Radio Layer – This layer moves the bits from master to slave or vice versa. It is the lower-power system with a range of 10 metres operating in the 2.4 GHz ISM band. The band is divided into 79 channels of 1 MHz each. Modulation is frequency shift keying, with 1 bit per Hz giving a gross data rate of 1 Mbps, but much of this spectrum is consumed by overhead. To allocate the channel fairly, frequency hoping spectrum is used with 1600 hops/sec and a dwell time of 625 µ sec.

Because both 802.11 and Bluetooth operate in the 2.4 GHz ISM band on the same 79 channels, they interfere with each other. Since Bluetooth hops are faster than 802.11, it is far more likely that a Bluetooth device will ruin 802.11 transmission than the other way around.

(ii)   Baseband Layer – The baseband layer is the closest thing Bluetooth has to a MAC sublayer. It turns the raw bit stream into frames and defines some key formats. This is traditional time division multiplexing, with the master getting half the slots and the slaves sharing the other half. Frames can be 1, 3 or 5 slots long.

Each frame is transmitted over a logical channel, called a *link*, between the master and a slave. Two kinds of links exist. The first is the *ACL (asynchronous connection-less)* link, which is used for packet-switched data available at irregular intervals. These data come from the L2CAP layer on the sending side and are delivered to the L2CAP layer on the receiving side. ACL traffic is delivered on a best-efforts basic.

The other is the *SCO (synchronous connection oriented)* link, for real-time data, such as telephone connections. This type of channel is allocated a fixed slot in each direction. Due to the time-critical nature of SCO links, frames sent over them are never retransmitted. Instead, forward error correction can be used to provide high reliability. A slave may have upto three SCO links with its master. Each SCO link can transmit one 64,000 bps PCM audio channel.

(iii)   Logic Link Control Adaption Protocol – It has three major functions. It accepts packets of up to 64 kB from the upper layers and breaks them into frames for transmission. At the far end, the frames are reassembled into packets again.

It also handles the multiplexing and demultiplexing of multiple packet sources. When a packet has been reassembled, the L2CAP layer determines which upper-layer protocol to hand it to.

L2CAP fulfils the quality of service requirements, both when links are established and during normal operation. Also negotiated at setup time is the

maximum payload size allowed, to prevent a large-packet device from drowning a small-packet device. This feature is needed because not all devices can handle the 64 kB maximum packet.

Q.43. Draw and explain the frame structure of Bluetooth protocol.

Ans. There are different frame formats of Bluetooth, one of them is shown in fig. 3.20. This starts with an access code that usually identifies the master so that slaves within radio range of two masters can tell which traffic is for them. Second field containing 54-bit header typical MAC sublayer fields. Finally, the data field of upto 2744 bits.
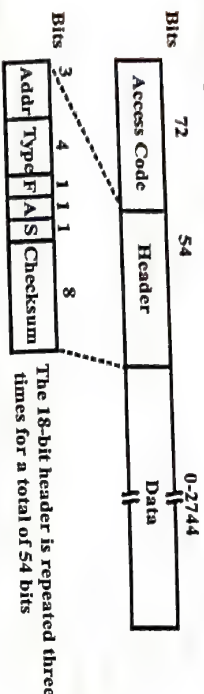

Bits 72   54   0-2744

| Access Code | Header | Data |

Bits 3   4   1 1 1   8

| Addr | Type | F | A | S | Checksum |

The 18-bit header is repeated three times for a total of 54 bits

Fig. 3.20 A Typical Bluetooth Data Frame

The header field itself contain various subfields. The *Address* field identifies which of the eight active devices the frame is intended for. The *Type* field defines the frame type (ACL, SCO, Poll, or null) the type of error correction used in the data field, and how many slots long the frame is. The *Flow* bit is asserted by a slave when its buffer is full and cannot receive any more data. This is a primitive form of flow control. The *Acknowledgement* bit is used to piggyback an ACK onto a frame. The *Sequence* bit is used to number the frames to detect retransmissions. After that, comes the 18-bit header Checksum. This 18-bit header repeated three times to form 54-bit header shown in fig. 3.20.

Various formats are used for the data field for ACL frames. The SCO frames are simpler though – the data field is always 240 bits. Three variants are defined, permitting 80, 160, or 240 bits of actual payload, with the rest being used for error correction. In the most reliable version (80-bit payload), the contents are just repeated three times, the same as the header.

Q.44. What is wireless sensor network ? Give its uses.

Ans. A wireless sensor network is made up of spatially distributed autonomous devices with sensors. These devices can monitor the environmental and physical conditions. A number of end-nodes, routers coordinated by a coordinator make a WSN. Several sensors are attached with each end-node. Data from end-nodes as communicated to the coordinator via routers in the form of data packets. Data from all the nodes is collected by the coordinator

which acts like gateway to connect the WSN with Internet. Some example of WSNs used in IoT system are given below –

   (i)   For monitoring the grid at various points, i.e. smart grid system, using WSNs.

   (ii) To monitor the health of structures like buildings and bridges by collecting the vibration data from structural health monitoring system using WSNs.

   (iii) For collecting surveillance data in a surveillance system using WSNs.

   (iv) To monitor the indoor air quality and concentration of various gases in indoor air quality monitoring system using WSNs.

   (v) To monitor soil moisture at various locations, in soil moisture monitoring system using WSNs.

Q.45. What are the salient features of WSN technology ?

Ans. In WSN every node sensor has the capability of computation, data compaction, aggregation and analysis. Each node also has communication and networking capabilities.

The power of WSNs lies in their ability to deploy greater number of low-cost and low-power sensing nodes. WSNs are self-organizing networks which makes them robust. The network can reconfigure itself when some new nodes are added in network or some nodes get failed.

Q.46. Explain wireless sensor network technology.

                                                     (R.G.P.V., May 2018, 2019)

Ans. Refer Q.44 and Q.45.

Q.47. Explain in brief context-based node operation.

Ans. The circumstances which form the setting of an idea, event or statement and in terms of which it can be fully understood is known as context. By using a sensor and associated circuit, computation and networking capabilities, and context at that node, a WSN node can adapt, re-program or do another work. The ability of nodes to change their operations according to changes in the environment is called context-based sensing, networking and computing, i.e. context based operations. The tasks that need to be undertaken on a changed context are identified by the application layer programs. In the WSN system, the selection of data, power, memory and routing path management, the routing protocol, users, devices and application interface can be programmed to function according to the context and consider the circumstances, while networking and computations.

In WSN nodes the context for re-programming may be in the form of user, structural, physical. Some examples of context are –

(i) Surrounding conditions in the past and at present.

(ii) Earlier cached data records.

(iii) Current network.

(iv) Currently available battery power and memory.

(v) Physical parameters (i.e. present time of day).

(vi) Nearest available connectivity.

(vii) Changes in the state of the connecting network.

(viii) Nearby systems or devices.

(ix) Applications sequence in the past.

(x) Device users past sequence of actions.

Q.48. Describe the various components of the WSN.

Ans. The components of WSN system are sensor node, relay node, actor node, cluster head, gateway and base station.

(i) Sensor Node – It is capable of executing data processing, data gathering and communicating with additional associated nodes in the network. A distinctive sensor node capability is about 4-8 MHz, having 4 KB of RAM, 128 KB flash and preferably 916 MHz of radio frequency.

(ii) Relay Node – It is a midway node used to communicate with the adjacent node. It is used to enhance the network reliability. A relay node is a special type of field device that does not have process sensor or control equipment and as such does not interface with the process itself. A distinctive relay node processor speed is about 8 MHz, having 8 KB of RAM, 128 KB flash and preferably 916 MHz of radio frequency.

(iii) Actor Node – It is a high end node used to perform and construct a decision depending upon the application requirements. Typically these nodes are resource rich devices which are outfitted with high quality processing capabilities, greater transmission powers and greater battery life. A distinctive actor node processor capability is about 8 MHz, having 16 KB of RAM, 128 KB flash and preferably 916 MHz of radio frequency.

(iv) Cluster Head – It is a high bandwidth sensing node used to perform data fusion and data aggregation functions in WSN. Based on the system requirements and applications, there will be more than one cluster head inside the cluster. A distinctive cluster head processor is about 4-8 MHz, having 512 KB of RAM, 4 MB flash and preferably 2.4 GHz of radio frequency. This node assumed to be highly reliable, secure and is trusted by all the nodes in the sensor network.

(v) Gateway – It is an interface between sensor networks and outside networks. Compared with the sensor node and cluster head the gateway node is most powerful in terms of program and data memory, the processor used, transceiver range and the possibility of expansion through external memory.

A distinctive gateway processor speed is about 16 MHz, having 512 KB of RAM, 32 MB flash and preferably 2.4 GHz of radio frequency.

(vi) Base Station – It is an extraordinary type of node having high computational energy and processing capability.

Q.49. What are the advantages and disadvantages of WSN ?

Ans. The advantages and disadvantages of WSN are as follows –

Advantages of WSN –

(i) It avoids a lot of wiring.

(ii) It can accommodate new devices at any time.

(iii) It is flexible to go through physical partitions.

(iv) It can be accessed through a centralized monitor.

Disadvantages of WSN –

(i) Lower speed compared to wired network.

(ii) Still costly at large.

(iii) More complex to configure than wired network.

(iv) It does not make sensing quantities in building easier.

(v) It does not reduce costs for installation.

(vi) Gets distracted by various elements like Bluetooth.

Q.50. Explain in brief architecture of WSN node.

Ans. WSN node has a three layer architecture as shown in fig. 3.21. Layers which are involved in this architecture are application layer, network layer, physical cum data-link layer. Sensor management, sensor query and data dissemination, task assignment and application-specific protocols are the components of the application layer software. The application and network layers have sensor, CPU and program sensor node. Data-link layer and network layer are serially connected and may have coordination of the routing software. Layers are interconnected to an antenna and wireless radio circuit by a serial link. The physical cum data-link layer contains the radio circuit, MAC and physical protocols are used by the communication subsystem.
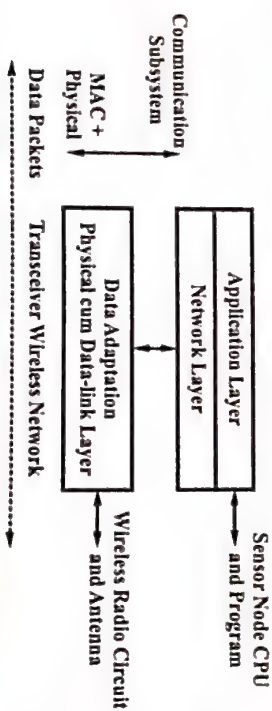

Fig. 3.21 Wireless Sensor Node Architecture

Q.51. Discuss architecture for connecting nodes in a WSN system.

Ans. Two commonly used architectures for connecting nodes in a WSN system as shown in fig. 3.22, are –

(i) Fixed connecting infrastructure of WSNs containing coordinators, relays, gateways and routers.

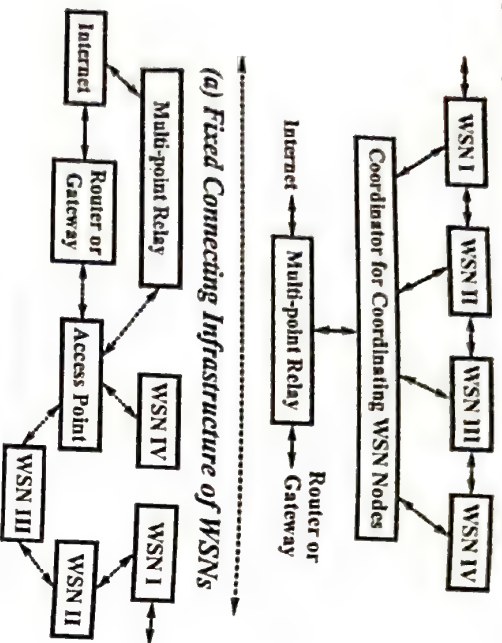(ii) Ad-hoc network of mobile WSNs containing coordinators, relays, gateways or routers.

(a) Fixed Connecting Infrastructure of WSNs

(b) Ad-hoc Network of Mobile WSNs

Fig. 3.22

The nodes present nearby or reaching in wireless range are provide accessibility by means of a fixed point transceiver known as an access point. To connect with other networks (i.e. mobile service provider network or internet), a multi-point relay is used. Two networks are linked together by a coordinator. The router selects the best path for transmission of data packet, among the available paths in a network at present. A smart home network having WSNs at security surveillance points, air conditioner, TV, computer with Wi-Fi access points is an example of fixed connecting infrastructure. Mobile WSNs tied with birds or animals for habitat monitoring is an example of ad-hoc infrastructure.

Q.52. Discuss architectures for networking of nodes with neat sketches.

Ans. Layered architecture and multiple cluster architecture are two basic architectures used for networking of WSNs. These architectures are discussed below –

(i) *Layered Architecture* – Wireless multihop infrastructure network architecture (MINA) is a layered architecture. The nodes of WSN have the

---

data sensing capability and can move towards the access point. These nodes are mobile and can move for communication to remote access points. The access point has the capabilities of data gathering and processing. It can connect with the larger network, link Internet.

Each node can connect with its nearby nodes, while node which moves to longer distances can communicate to the access point through 2 or 3 hops. Each node also has a low power transceiver to connect with the neighbouring layer WSNs.

In a layered architecture as shown in fig. 3.23 the access point is surrounded by three layers of WSNs. Layer 1 WSNs are directly connected with the access point. While layer 2 WSNs are first connected to layer 1 WSNs functioning like coordinators and then directly connected to the access point. Similarly, layer 3 WSNs first connect to layer 2 WSNs, then to layer 1 WSNs and then to the access point.
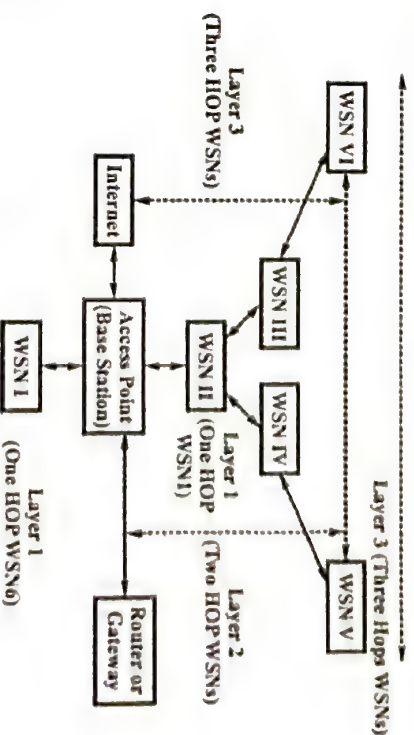

Fig. 3.23 Layered Architecture

As shown in fig. 3.23 WSN I and WSN II of layer 1 are directly connect to the access point, thus have hop count = 1. While WSN III and WSN IV of layer 2 are first connected by one hop to WSN II and next hop to access point, thus hop count = 2. In a similar way, WSN V and WSN VI which are at layer 3, will connect to the base station by 3 hops. With the help of wireless LAN protocol (802.11b) base stations connect with the clusters. The access points also have connectivity to the Internet. Sensor data is archived and can be queried, in real time. This data can be accessed by users through mobile devices and remote clients.

(ii) *Multiple Cluster Architecture* – A cluster refers to a group of WSNs. Each cluster has an associated gateway node. A set of such clusters, each of which having a gateway, has one cluster with a cluster-head gateway.

In a multiple cluster architecture, cluster-head allows a tree-like topology of the clusters. In distributed WSNs and WSN clusters, the formation of clusters and election of cluster head is autonomous. A multiple cluster architecture as shown in fig. 3.24, has two clusters and a cluster-head gateway. Head gateway connects to the Internet to provide connectivity to WSNs in the multiple clusters. The number of clusters required in a system, depends upon the coverage required. Two clusters are interconnected by the gateway of clusterhead. Each node in a cluster is connected to its nearby node. WSNs of two clusters interconnect via gateway in one, two or three hops. A node which moves to longer distances, communicates the neighbouring cluster through the gateway.
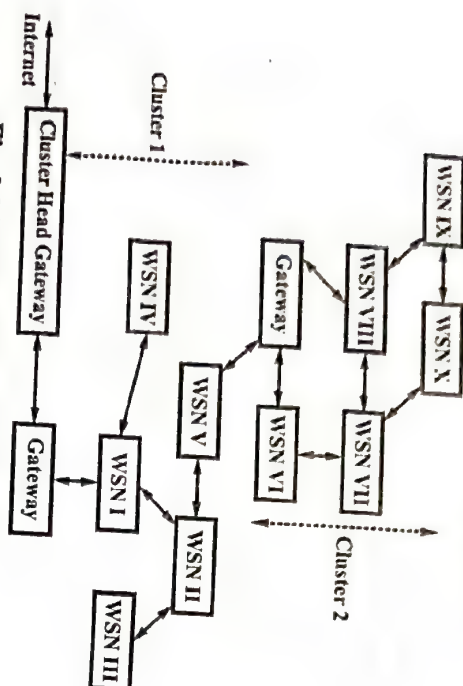

Fig. 3.24 Multiple Cluster Architecture

In the architecture shown in fig. 3.24, clusters 1 and 2 are ad-hoc networks of mobile WSNs, and each of these clusters may have layered or mesh architecture. Data compaction or fusion and aggregation are allowed by clustered architecture. Such compacted or aggregated data is communicated by the clustered to another cluster. Before communicating this data to a web server or cloud over Internet, a cluster-head further aggregates, compacts or fuses the data.

Q.53. Write in brief about WSN protocols.

Ans. WSN protocols are designed to –

(i) Decrease computational requirement.
(ii) Reduce use of bandwidth and battery power.
(iii) Operate in self-configured ad-hoc setup mode.
(iv) Reduce memory requirement by the protocol.

The physical layer of WSN uses adaptive RF power control. It increases the power, when nearest node is far away and reduces power when node is nearby node. It uses CMOS low power ASIC circuits and energy efficient codes.

Data-link Layer Media Access Control (MAC) Protocol – At the data link layer, an S-MAC protocol can be used. The S-MAC nodes go to sleep for extended periods and need to synchronise at periodic intervals. Use of S-MAC protocols results in energy efficient, collision-free transmission, and intermittent synchronisation of the operations. Collision-free transmission occurs due to scheduling of channel which also allows reuse of channels.

Routing Protocols – Routing protocols are needed for energy-efficient routing, multi-hop route determination, route caching and directed diffusion of data by network layer.

Routing protocols can be of two types, viz. reactive or proactive. *Reactive* protocols are used to determine the route on demand, while *proactive* protocols keep route cache determine the route in advance. When a routing table guides the paths available, routing protocols are table-driven. When a source demands the route and guides the paths available, routing protocols are demand-driven.

Q.54. What points are need to be considered for WSN infrastructure establishment ?

Ans. The following steps need to be considered, while establishing a WSN infrastructure –

(i) Sensors and their CMOS low power ASIC circuit with sensors along with their radio ranges and energy-efficient coding.
(ii) Clusters and their hierarchy, clusters gateways, cluster-heads.
(iii) Routing, aggregation, compaction, fusion and diffusion of data.
(iv) Network topology and architecture which may be either wireless multi-hop infrastructure network architecture or multiple cluster architecture, depending upon the type of applications and services.
(v) Time should be synchronised at intermittent intervals.
(vi) Selection of infrastructure either –

(a) fixed connecting infrastructure of nodes, coordinators, relays, gateways and routers, or

(b) ad-hoc network of mobile WSNs with limited or unspecified mobility region.

(vii) Self-discovering, self-configuring and self-healing protocols of network nodes, localisation, mobility range, security, data-link and routing protocols, link quality indicator, QoS required according to applications and services.

Q.55. Why WSN need integration approaches ?

Ans. WSN requires integrated approaches for the following purposes –

(i) Nodes design and provisioning of resources
(ii) Mobility of nodes

(iii) Localisation of nodes

(iv) Architecture for connecting sensors

(v) Architectures for sensor networking

(vi) Security protocols

(vii) Data dissemination protocols

(viii) Protocols of data-link layer and routing

(ix) For IoT applications and services, integration with sensors data.

Q.56. What are the security challenges with the WSN networks ?

Ans. Security challenges for WSN networks are –

(i) An attacker node drains out the energy of the attacked node by repeatedly sending hello messages, known as **hello flood attack.**

(ii) An attacked node acts like an access point and receives the messages without forwarding them, known as **sinkhole attack.**

(iii) The attacker node provides wrong information about the distances of the destinations, hence forcing the attacked node to take longer paths having high latency which results in high delivery delays of packets. This type of attack is known as **wormhole attack.**

(iv) A single node, presents itself as different entities at different times, known as **sybil attack.**

(v) The attacker node does not forward the messages received from an attacked node, known as **selective forwarding attack.**

Q.57. Write short note on quality of service.

Ans. Quality of Service (QoS) is an average weighted metric measured over the life time of a WSN network. It depends upon following parameters –

(i) Maximum time up to which the WSN functions effectively or given energy resources of the nodes will last i.e., life time.

(ii) Time taken in generation of sensor data and its delivery up to the destination i.e., average delay.

(iii) Bytes per second delivered upto the destination i.e., throughput. High delays means low throughput. It also related to the bandwidth of the network.

(iv) Link Quality Indicator (LQI), which is a measure of packets delivered/transmitted from the nodes.

Some factors which limit high QoS are –

(i) Routing through nodes having high throughputs, i.e. lower delay and paths using low energy resources.

(ii) Link between nodes.

(iii) Maintaining a balance between priority and delay. Higher priority packets should be delivered by lower delay paths and lower priority packets by higher delay paths.

An example of QoS metric is the coverage of a sensor network, which depends on the density and locations of the nodes in the region, their communication range and sensitivity. Another QoS metric, which is used for surveillance systems, hazardous gases or fire-detection systems is percentage of times the network cover occurrence of an event.

Q.58. List the design challenges of configuring of nodes.

Ans. Challenges in configuration of nodes (which may be static, dynamic or self-automatic) with respect to the resource constraints are –

(i) WSN nodes in locations and mobility range.

(ii) Clusters

(iii) Gateways

(iv) Cluster-heads

(v) Sampling rate of the sensed parameters

(vi) Aggregation, compaction and fusion.

Resource constraints in view of which these challenges are to be met include limited battery power, computation capability, storage, bandwidth and scalability limit on the nodes.

Q.59. What are the protocols used for secure communication between WSN nodes.

Ans. Data privacy and integrity, are two major requirements for secure communications between sensor networks. Data integrity can be maintained by receiving the data from authenticated sensing nodes only and preventing the communication of messages from unauthenticated sources. Privacy ensures secrecy of data and prevents eavesdropping. SPINS (Security Protocols in Network of Sensors) suggested by Berkeley laboratories is the best protocol for secure communication between networks. SPINs use symmetric cryptographic protocol because of the following reasons –

Symmetric cryptographic method has low overhead in term of memory and computations for digital signatures, key generation and verification as compared to asymmetric method which has high memory needs and communicates higher number of bytes.

SPINS is actually a suite of security protocols including –

(i) SNEP (Secure Network Encryption Protocol)

(ii) Micro-Tesla (μ-Tesla).

SNEP ensures data integrity and privacy, thus enables secure point-to-point communication. In this protocol, messages are not need to be replayed, thus they remain fresh. The communication requires an authentication process. In SNEP, access point distributes session key to two nodes A and B. Total six keys are shared by two nodes viz., encryption keys, K_{AB} and K_{BA}, Cipher-

block-chaining message authentication code keys, K_{AB} and K_{BA} and counter keys, C_A and C_B.

A lighter version of TESLA known as μ-TESLA enables authenticated broadcasting. The authentication process is micro-timed and efficient. First a packet is listened and considered as a parent and authenticated later. This results in stream-loss tolerant secure authentication.

Q.60. Explain in brief localised encryption and authentication protocol.

Ans. In a distributed sensor network, a node uses four types of keys, viz. individual key, group key, cluster key and share a pair of key with the neighbour. In Localised Encryption and Authentication Protocol (LEAP), different packets use different keying mechanisms, depending upon the security needs of the networks. The implementation of goals of a system for applications and services is a problem for highly-distributed architecture with localised coordination.

Various implementation requirements include autonomous operation, self-organization, self-configuration, adaptation, energy conservation at physical, MAC, link, route, application layer, design of scalable node density, number and type of networks.

Route nodes have no addressability and network is a data-centric network.

Q.61. What are the application of WSN ?

Ans. WSN applications are as follows –

(i) In Agriculture – For humidity/temperature monitoring.

(ii) In Civil Engineering – For structural response and disaster management.

(iii) In Environment Sciences – For habitat monitoring and conservation of biology.

(iv) In Home and Office Applications – For home and office automation and smart environment.

(v) In Health Applications – For telemonitoring of human physiological data and drug administration in hospitals.

●●

MQTT, MQTT METHODS AND COMPONENTS, MQTT COMMUNICATION, TOPICS AND APPLICATIONS, SMQTT

Q.1. What is MQTT and give its advantages ?

Ans. The full form of MQTT is "Message Queuing Telemetry Transport". It is an open-source protocol. It is used for M2M or Internet of Things connectivity. It was introduced first by IBM and then IBM donated it to M2M 'Paho' project of Eclipse. A version represents MQTT v3.1.1. It has accepted as organisation for the advancement of structured information standards (OASIS) standard in year 2014. In M2M or IoT devices MQTT protocol is
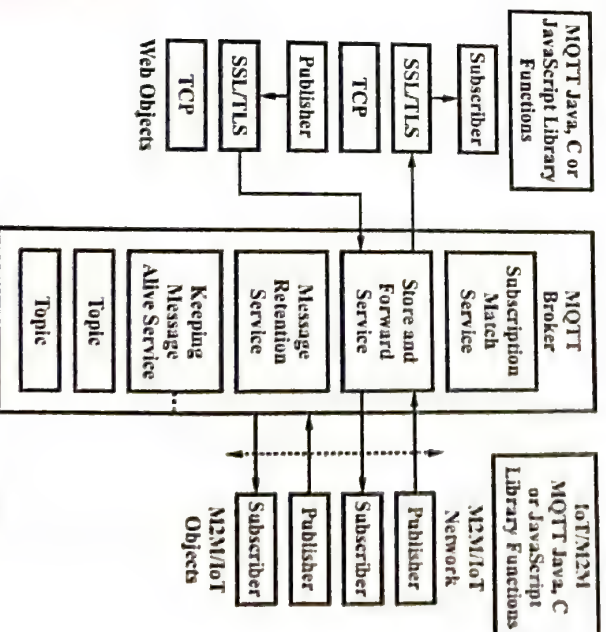

Fig. 4.1

used for connectivity. A version represents MQTT-SN v1.2. MQTT-SN is used in sensor networks and non-TCP/IP networks like ZigBee. Messaging protocol is also publish/subscribed by MQTT-SN. It allows extension of the MQTT protocol for WSNs, the actuator and sensor devices and their networks.

MQTT-broker subscription, store and forward, subscription match, last good message retention and keep messages alive services are shown in fig. 4.1. Here, device objects employ MQTT Java, C or JavaScript library functions. With the help of an MQTT broker, messages interchange between M2M/IoT device object and web objects. The object communicate with the help of connected devices network protocols like ZigBee. In addition, web objects employ MQTT library functions and communicate with the help of IP network and TLS and SSL security protocols in case of publishing and subscribing web APIs.

The following functions are performed by MQTT brokers –

(i) Store-and-forward function is performed. It stores topics from publishers and transmits to subscribers.

(ii) Functions like server node has ability to store message from publishers, and transmitting them to the subscribing clients.

(iii) It has ability to receive topics by the publishers. These topics are used to measure like measured information of ambient light conditions, nearby parking space availability and traffic density waste container status.

(iv) Until the client explicitly disconnected, recovers subscriptions on reconnect after a disconnection.

(v) In order to route messages to right end points, receives subscriptions from clients on the topics, matches these subscriptions to publications.

(vi) Between publisher and subscribers of the topics, it behaves like broker.

(vii) Authentication with the help of username/password for connect message and client security represents with the help of SSL/TLS.

(viii) With the help of Intelligent and business analyst server and other servers MQTT server can be used with a gateway.

Advantages of MQTT –

(i) Small code footprint

(ii) Low network bandwidth requirement

(iii) Faster response time

(iv) Low power requirement

(v) Ease of scalability.

Q.2. Write short note on MQTT methods and components.

Ans. In MQTT there are a few different methods such as connect, disconnect, subscribe, unsubscribe and publish. Basically the connect method helps to connect this device with the server. The disconnect is just opposite. Whenever a device is no longer required to be to remain connected, the disconnect method helps in disconnecting from the server, from TCP IP service offerings and so on. The subscribe method is basically for subscribing to the services and unsubscribe is the opposite which is used when it is no longer required to continue with getting the different data offerings, the data services and so on.

The publish method which is basically publishing data from the different sensors or the different devices to the broker for it to be fetched by the different application clients.

The MQTT has three principle components. The publishers which involve the different sensors, the subscribers and that means, those entities, those applications, those units that are interested in the data that is published by the sensors. In between the two is the broker which helps the publishers and the subscribers connect to one another and also help in classifying the sensor data into different topics.

Q.3. Explain in detail about the architecture of MQTT.

Ans. The MQTT consists of three components, subscribers, publishers, and brokers. The architecture of MQTT is shown in fig. 4.2. To communicate, the device will register as a subscriber to a specific topic of its interest in the broker. When the publishers publish to the topic, the broker delivers the information to one or more subscribers. Many applications use MQTT e.g., healthcare, monitoring, energy meter, facebook notification. Furthermore, the MQTT represents one of the appropriate messaging protocol for IoT and M2M communications because provides routing for small, cheap, limited power and memory devices that belong to vulnerable and low bandwidth networks.

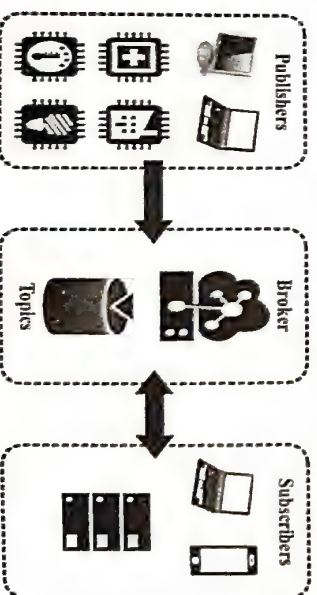Publishers — Topics — Broker — Subscribers

Fig. 4.2

The MQTT implements the communication between the broker and the publishers and the broker with the subscribers. To start a communication, CONNECT packet is sent from the client (publisher or subscriber) to the server (broker) to start the connection, the server responses with a CONNACK packet. To finish the communication, the client sends a DISCONNECT packet.

Q.4. Discuss communication process in MQTT.

Ans. During the communication process following three Quality of Service (QoS) levels may be used in MQTT –

(i) QoS 0 (At most Once) – The message is sent using the best effort on the TCP/IP network, the answer is not expected and the message is not sent again. The message can arrive at the server or not. The sender sends a PUBLISH packet as shown in fig. 4.3.
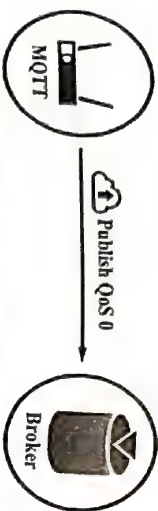

Fig. 4.3 QoS 0 (At Most Once)

(ii) QoS 1 (At Least Once) – It is the default mode of message transfer. The message arrives at least once to the receiver, to ensure delivery at least once. If a failure is identified, there may be failure on the communication link, or the message is not received after a specified period of time, the message will be resent by the sender. It allows the recipient to receive the message multiple times. After the message is processed, it is deleted from the receiver. The sender using QoS 1 sends a PUBLISH packet containing the Packet identifier. The Publish packet retains a state of unacknowledged until the sender, receives the PUBACK packet from the broker/receiver. Fig. 4.4 shows message exchange between a publisher and broker using QoS 1 technique. After deleting the message the receiver sends the acknowledgement to the sender. The same occurs with the sender after receiving the acknowledgement from the receiver. Both sender and receiver delete the message after the communication.
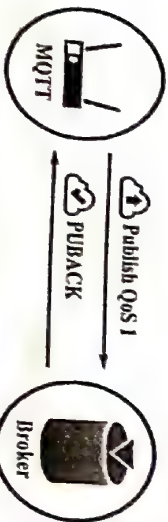

Fig. 4.4 QoS 1 (At Least Once)

(iii) QoS 2 (Exactly Once) – Duplicate messages and losses are not acceptable. It increases the network traffic, but it is acceptable because the QoS 2 is used to send most messages, it is the highest quality of service.

This technique guarantees that a message is received once, when the receiver confirms the message has arrived. The QoS 2 PUBLISH packet involves two-step process. The PUBLISH packet is treated as unacknowledged until the sender receives the corresponding PUBREC packet from the receiver. The sender sends a PUBREL packet and waits for the corresponding PUBCOMP packet from the receiver. The PUBREL packet is treated as unacknowledged until the PUBCOMP packet reception. Fig. 4.5 shows messages exchange between a publisher and broker using QoS 2 technique.
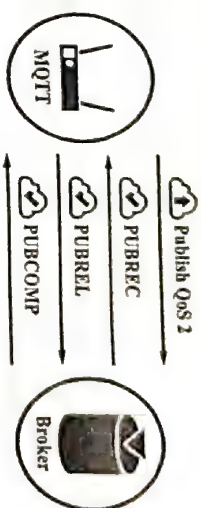
Fig. 4.5 QoS 2 (Exactly Once)

The MQTT specification do not present parameters to be set in order to define an interval, and how many times the messages may be retransmitted.

Q.5. Give features and limitations of commonly used brokers in MQTT.

Ans. The MQTT broker is the heart of each MQTT arrangement. It provides connecting link between applications or physical devices and enterprise systems. Brokers are in charge of subscription, determined sessions, missed messages and general security, including authentication and authorization. The following table 4.1 gives the commonly used MQTT brokers with their features and limitations.

Table 4.1 MQTT Broker with their Features and Limitation

S.No.	Broker	About	Features	Limitations
(i)	Mosquitto	It supports MQTT version 3.1 and it is an open source.	· All QoS · Authentication · Bridge · Dynamic topics · Websockets	· Clustering · Fewer Configuration Now allow simultaneous connection with using authentication
(ii)	RSMB (Really small message broker)	It is a tiny broker which supports V3 and V3.1.	· All QoS · Bridge · Dynamic topics	· Security · Websockets · Cluster

(iii)	MQTT.js	It is an MQTT broker among client/server API production recorded in JavaScript.	· All QoS · Dynamic topics · Websockets	· Bridge · Authentication · Cluster
(iv)	HiveMQ	HiveMQ empowers organization to attach all devices and services with nominal effort by victimization the de-facto.	· All QoS · Bridge · Dynamic topics · Websockets · Cluster	· SSL · Open standard · Performance degradation because of TLS/SSL TLS
(v)	VerneMQ	VerneMQ cloud be a superior, distributed MQTT message broker supports MQTT version 3.1 and 3.1.1.	· All QoS · Bridge · Dynamic topics · Websockets · Encryption	· Authentication · Performance degradation because of TLS

Q.6. Define the term MQTT topics.

Ans. A topic is a simple string that can have more hierarchy levels, which are separated by a slash. A sample topic for sending temperature data of the living room could be *house/living-room/temperature*. On one hand the client (e.g. mobile device) can subscribe to the exact topic or on the other hand, it can use a wildcard. Each client that wants to receive the messages subscribes to a certain topic and the broker delivers all the messages within the matching topic to the client. Thus, clients, do not have to know each other they only need to communicate with each other over the topic. This architecture basically appears to be a scalable architecture with a scalable solution. There is not much dependency between the producers and the consumers of the data and that is why MQTT is very popular.

The subscription to *house/+/temperature* would result in all messages sent to the previously mentioned topic *house/living-room/temperature*, as well as any topic with an arbitrary value in the place of living room such as *house/ kitchen/temperature.*

The plus sign is a single level wild card and only allows arbitrary values for one hierarchy. If more than one level needs to be subscribed, such as, the entire sub-tree, there is also a multilevel wildcard (#). It allows to subscribe to all underlying hierarchy levels. For example *house/#* is subscribing to all topics beginning with *house.*

Q.7. Explain the applications of MQTT in various fields.

Ans. The applications of MQTT are as follows –

(i) Healthcare – By using MQTT, a healthcare association needed to create a flexible checking solution. Following are the arrangement expected to address of victim care –

    (a) Keeping track of victims besides they go away from the clinic.

    (b) Upgrading the effectiveness of subsequent tests.

    (c) Achieving advanced industry information catch principles.

The healthcare organization worked with IBM to make an answer in which an MQTT customer is inserted in a home observing machine that gathers diagnostics at whatever point the victim is in nearness to a base system. Then it forwards the indicative information through the web to the main domain, which is given to an application which analyses the measurements and alert the healthcare team if there are hints the victim perhaps carrying trouble. It spares cash for the association also its victims, as there is constrained requirement for victims to go hospital for regular check-ups if they are doing fine.

(ii) Energy and Utilities – A service organization was confronted with increasing expenses to deliver power among with rising interest for electricity by their client root, which was not able, normally, to spend forever expanding amounts. Therefore instead of quickly carried out generation charges that their clients possibly could not spend, the organization first looked for an answer for decrease general request for power by putting smart meters in clients' apartments to remotely manage the application of definite power absorbing device. In any case, the arrangement expected to minimize utilization of accessible information network, for that the organization salaried according to the quantity of information transferred.

Making of virtual power plant (VPP) was the arrangement which sits between the organization's producing origins and their clients. Smart meters gather use information for the different devices which are used there in-home apartment. At that point, apartment gateway examines, furnished with a progressed MQTT customer, distribute the utilization information to the VPP at normal interval throughout the nearby cell phone system.

(iii) Social Networking – A long range interpersonal communication organization experienced latency issues during transferring information. The strategy the organization utilized to deliver data was stable yet time-consuming, and if it remained to utilize the similar mechanism then the solutions were restricted. Another structure for a constant association among the servers lacking absorbing more battery power was required that is basic to clients of the organization's civic communication site.

Using MQTT protocol the organization's designers tackled the issue in social networking. With keeping up a MQTT association also directing data via MQTT's conversational channel (chat pipelin), the organization was capable to accomplish data distribution by speeds of 1×105 microseconds, instead of several minutes.

Q.8. What are the limitations of MQTT ? Explain.

Ans. MQTT is the most popular communication protocol for IoT since IoT assets are constrained devices. But it has following limitations which need to be addressed –

(i) Message Expiry – Right now in MQTT there is no message expiry, so in the event that you put a message in a broker and then forget to gather it or nobody ever comes to pick it up, it remains there forever. As a result, the broker is overloaded with messages and it degrades the overall performance.

(ii) Security – MQTT protocol provides username and password for authentication and various broker implementations add different mechanism on top of that. So security in MQTT depends on the use case and selection of broker. Mostly brokers provide security based on TLS, but TLS affects the performance significantly, especially CPU usage during the handshake.

(iii) Ordering – The key challenges for a reliable data transmission organization in IoT environment are ordering messages and resending messages which lost during transmission. MQTT provides guaranteed delivery of messages, but maintaining order of messages in MQTT is a challenging task.

(iv) Priority – MQTT does not support a feature called priority of messages. If any system has more important data then it must be immediately available to all the subscribers, for example the data which is gathered from fire alarm system is more important than temperature or pressure sensor data so it must be available first to all receivers. So for that priority of messages are required then sending data in order.

Q.9. Write short note on SMQTT.

Ans. The secured version of MQTT which is called the secure MQTT, or SMQTT in short. This is quite similar in notion to http and https, the secure http. So, secure MQTT is an extension of the MQTT using different security features such as encryption and so on. The advantage of such encryption is the broadcast encryption feature in which one message is encrypted and delivered to multiple other nodes which is quite common in IoT applications. In general, the algorithm consists of four main stages, i.e. the setup stage, the encryption stage, the publish stage and the decryption stage. In the setup phase, the subscribers and publishers register themselves to the broker and

get a master secret key according to their developer's choice of key generation algorithm. When the data is published, it is encrypted and published by the broker which sends it to the subscribers, which is finally decrypted at the subscriber end and having the same master secret key. The key generation and encryption algorithms are not standardized. SMQTT is proposed only to enhance MQTT security features.

Q.10. What do you mean by MQTT-SN ? Explain.

Ans. The Message Queuing Telemetry Transport for Sensor Network (MQTT-SN) protocol was developed specially for Wireless Sensor Networks (WSNs), normally made up of low cost and easy developed environments. Typically, a WSN has a large number of sensors and actuators, from different device types, that present a limited amount of storage and processing capabilities. The devices are developed to detect and notify events through wireless links, used habitually in monitoring environment, traffic and building management, battlefield surveillance, and home automation.

MQTT-SN is also optimized for implementation on low-cost, battery-operated devices with limited processing and storage resources.

MQTT-SN was originally developed for running on top of the ZigBee APS layer. ZigBee is an open industrial consortium with the aim of defining an open and global communication standard for WSNs. To be global ZigBee has selected the IEEE standard 802.15.4 as the protocol for the PHY and MAC layers, and adds on top on this standard the required network, security and application layers, thus providing interoperability between products of different vendors.

MQTT-SN is designed in such a way that it is agnostic of the underlying networking services. Any network which provides a bi-directional data transfer service between any node and a particular one (a gateway) should be able to support MQTT-SN. For example, a simple datagram service which allows a source endpoint to send a data message to a specific destination endpoint should be sufficient. A broadcast data transfer service is only required if the gateway discovery procedure is employed. To reduce the broadcast traffic created by the discovery procedure, it is desirable that MQTT-SN could indicate the required broadcast radius to the underlying layer.

Q.11. Describe the architecture of MQTT-SN protocol.

Ans. The architecture of MQTT-SN is shown in fig. 4.6. There are three kinds of MQTT-SN components – MQTT-SN clients, MQTT-SN gateways (GW), and MQTT-SN forwarders. MQTT-SN clients connect themselves to a MQTT server via a MQTT-SN gateway using the MQTT-SN protocol. A MQTT-SN gateway may or may not be integrated with a MQTT server. In

case of a stand-alone gateway, the MQTT protocol is used between the MQTT server and the MQTT-SN gateway. Its main function is the translation between MQTT and MQTT-SN.

MQTT-SN clients can also access a gateway via a forwarder in case the gateway is not directly attached to their network. The forwarder simply encapsulates the MQTT-SN frames it receives on the wireless side and forwards them unchanged to the gateway in the opposite direction, it decapsulates the frames it receives from the gateway and sends them to the clients, unchanged too.
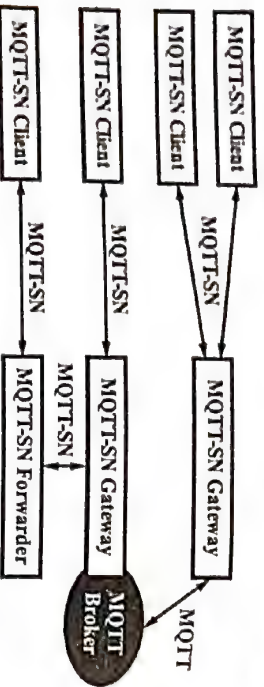

Fig. 4.6 MQTT-SN Architecture

Depending on how a gateway performs the protocol translation between MQTT and MQTT-SN, there can be two types of gateways, namely transparent and aggregating gateways, see fig. 4.7. They are explained in the following sections –

(i) Transparent Gateway – For each connected MQTT-SN client, a transparent gateway will setup and maintain a MQTT connection to the MQTT server. This MQTT connection is reserved exclusively for the end-to-end and almost transparent message exchange between the client and the server. There will be as many MQTT connections between the gateway and the server as MQTT-SN clients connected to the gateway. The transparent gateway will perform a "syntax" translation between the two protocols. Since all message exchanges are end-to-end between the MQTT-SN client and the MQTT server, all functions and features that are implemented by the server can be offered to the client.
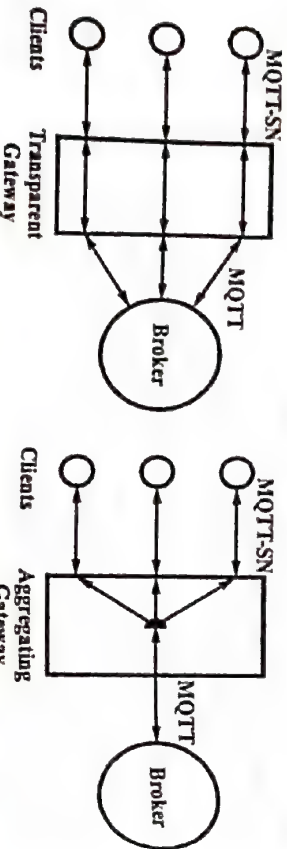

Fig. 4.7 Transparent and Aggregating Gateways

Although the implementation of the transparent gateway is simpler when compared to the one of an aggregating gateway, it requires the MQTT server to support a separate connection for each active client. Some MQTT server implementations might impose a limitation on the number of concurrent connections that they support.

(ii) Aggregating Gateway – Instead of having a MQTT connection for each connected client, an aggregating gateway will have only one MQTT connection to the server. All message exchanges between a MQTT-SN client and an aggregating gateway end at the gateway. The gateway then decides which information will be given further to the server. Although its implementation is more complex than the one of a transparent gateway, an aggregating gateway may be helpful in case of WSNs with very large number of SAs because it reduces the number of MQTT connections that the server has to support concurrently.

Q.12. Explain the following –
(i) MQTT-protocol architecture
(ii) MQTT-SN protocol architecture.

Ans. Refer to Q.9 and Q.11.

Q.13. Explain internet based communication and list its different kinds of protocols. (R.G.P.V., Nov. 2019)

Ans. The actions which occur during the data transferring from one layer to another layer are as follows –

(i)   For communication, every layer processes the data according to protocol used by that layer.

(ii)   The data stack sends and obtained from the previous upper layer with addition a new header it transmits by each layer and hence made a new stack after doing the actions defined at that layer.

(iii)   As per protocols, another layer will define the new parameters and make new stack for the subsequent lower layer.

(iv)   The process continues till the data communicates over the whole network.

In the modified OSI model, IoT physical and application layers are the lowest layer and highest layer respectively. When data transfers from the application layer to the physical layer, i.e. from one layer to the another layer following actions are taken –

(i)   Every layer works the processing according to the header field bits, and one taken as per the protocol to be employed for decoding the field for the needed actions at that layer.

(ii) Every layer takes the data stack from preceding lower layer and after the needed actions, it detects the header word and generates a new stack defined for the next higher layer.

(iii) The process continues till the data on the highest application layer is received at the port.

Lower layer such as data-link layer protocol has provisions for the trailing bits with header words, while upper layer uses only the header words. Training bits usage can be as end-of-the frame showing bits and error-control bits. TCP/IP protocol suite for Internet communication has only four layers of OSI model that are 7, 4, 3 and 2. Layer 1 is used for physical link to routers. Communication between the source and destination is shown in fig. 4.8. Link layer L2, Internet layer L3, transport layer L4 and application layer L7 are used by Internet-based TCP/IP communication. Protocol data units at the layers are shown in fig. 4.8. For transport layer TCP, a data segment is accepted from L7 layer. Then TCP stream is created by L4 layer. Stream is packetizes in the Internet layer L3. Although, for transport layer UDP usage, a datagram is accepted from L7 layer. Then UDP datagram is created by L4 layer. The stream packetizes into packet at the Internet layer L3. Datagram sent from layer L3 has maximum size of 2^{16} byte but it includes L3 header also.

Fig. 4.8
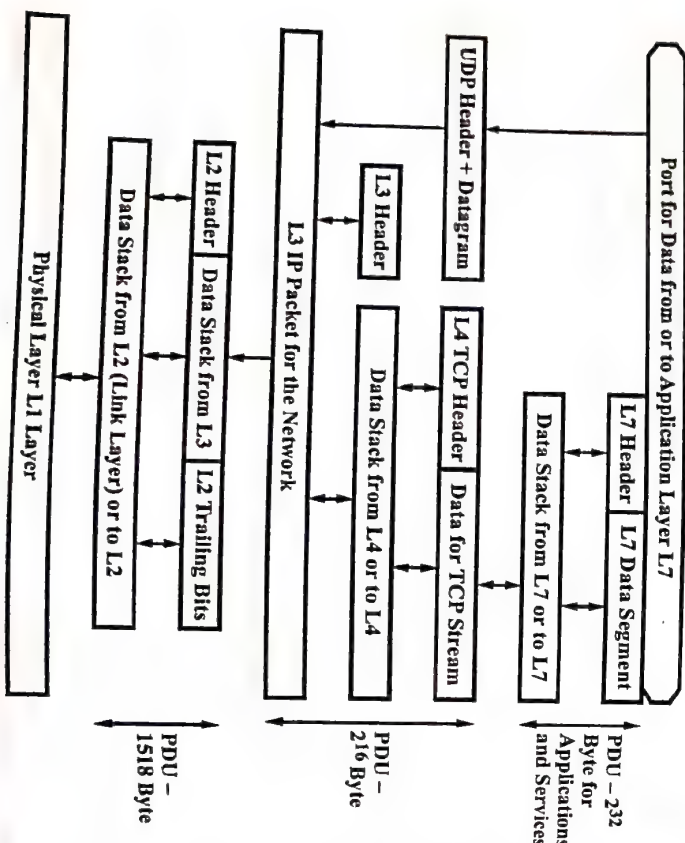
IP protocol is used by Internet layer. Packet routing obtains as – every router has knowledge about the path to destination. The multiple packets of the same source similarly follows various paths from a router, if the multiple paths are available. In source transport layer data streams, the destination-end transport layer is rearrange the packet as per to their orders. After rearranging the order of data segment it is send to destination-end of IoT application layer. Protocol is used by data link layer for its every sublayers such as RARP, PPP, NDP, Ethernet IEEE 802.3, ARP, MAC, etc. In logical link layer, Ethernet is a protocol on a LAN. The meaning of different terms used for protocols are RARP (Reverse Address Resolution Protocol), PPP (Point-to-Point Protocol), ARP (Address Resolution Protocol), MAC (Media Access Control), NDP (Network Discovery Protocol). Resolution stands for to use network data stack for computing the MAC address. Reverse stands for to use MAC address for computing IP address.

The following communication protocols have immediate importance to consumer and industrial IoTs –

(i) IEEE 802.15.4   (ii) ZigBee   (iii) 6LoWPAN
(iv) Wireless HART   (v) Z-wave   (vi) ISA 100
(vii) Bluetooth   (viii) NFC   (ix) RFID

CoAP, CoAP MESSAGE TYPES, CoAP REQUEST-RESPONSE MODEL

Q.14. Write short note on CoAP.

Ans. The constrained application protocol (CoAP) is an application layer protocol developed by the Internet Engineering Task Force (IETF) CoRE working group. It is designed for constrained environments. Based on a REST style architecture, the protocol considers the various objects in the network as resources. A unique Universal Resource Identifier (URI) is assigned to each resource. The protocol uses the corresponding URI to operate the different resources.

CoAP protocol was designed to look like and be compatible with hypertext transfer protocol (HTTP). This makes the protocol looks more like a traditional website-based business which provides the ability to be compatible with an existing system that is web service based.

Q.15. Describe the architecture of CoAP protocol.

Ans. The CoAP interaction model is similar to HTTP client/server model but the CoAP implementation acts as both client and server in typical machine to machine interactions. Similar to HTTP a CoAP request is sent by a client using a method code to request an action on a URI identifiable resource.

The server replies with a response code which may include a resource representation. CoAP model is essentially a client/server architecture enabling the client to request for service from server as needed and the server responds. CoAP request is similar in nature to HTTP but as shown in fig. 4.9 CoAP interchanges are asynchronous since it uses UDP. The message layer interfaces with UDP layer which formats the data received into a datagram and sends it to the lower layers of the OSI or the TCP/IP model.

| Application |
| CoAP Requests/Responses |
| CoAP Message |
| UDP |

Fig. 4.9 CoAP Architecture
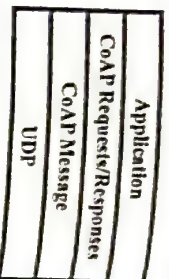
Similarly in the opposite direction, the datagram received by UDP presents it to the application layer in a legible format. Logically CoAP comprises of two-layers – a message layer responsible for UDP communication and reliability (optional), while the other layer is responsible for request or response interactions. CoAP also uses asynchronous message exchange between end points. CoAP defines four types of messages –
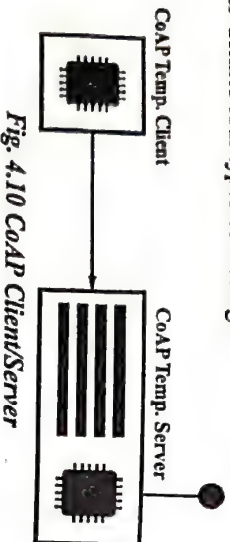

Fig. 4.10 CoAP Client/Server

CoAP defines four types of messages as Confirmable, Non-confirmable, Acknowledgement and Reset. The embedded method codes and response codes in some of these messages mark them as requests or responses. The code field of the CoAP header defines four message types. The example of CoAP client/server model, is shown in fig. 4.10 where a temperature sensor is installed in the conference room of an office. The temperature sensor works like a "server" (Thing) which any CoAP based client (another Thing) can query to get the temperature.

Q.16. Discuss the various message types used in CoAP.

Ans. The CoAP messages are similar to request or response model used in HTTP. The code field of the CoAP header defines four message types. The requests are carried in confirmable and non-confirmable messages whereas responses in these as well as piggybacked in acknowledgement messages. These message types are as follows –

(i) Confirmable (CON) – This message type requires response that is once message is sent; the receiver must confirm the message receipt.

(ii) Non-confirmable (NON) – This message does not require response that is once message is sent, the receiver does not need to confirm the message receipt. Thus implying unreliable message type.

(iii) Acknowledgement (ACK) – This message type is received in response to CON message confirming the latter reception.

(iv) Reset (RST) – This message is sent in case of an error in message, response is not understandable or receiver is not interested in communication with sender.

Q.17. Discuss various fields in CoAP message format.

Ans. A binary encoded CoAP message contains a CoAP header, options and payload. The header decides the available in a Type Length Value (TLV) format, as shown in fig. 4.11. The message fields are defined as follows with all lengths in unsigned integer formats –

(i) Version (Ver) – The field length is two-bit showing the CoAP version number, which is one at present.

(ii) Type (T) – This field is two-bit long showing the message type. There are four message types confirmable, non-confirmable, acknowledgement and reset represented by bit patterns of 00, 01, 10 and 11 respectively.

(iii) Option Count (OC) – It is four-bit long field, thus providing maximum of sixteen options after the header.

(iv) Code – It is eight-bit long field which shows whether the message is empty (0), request (1-31) and response (64-191). The remaining (192-255) is reserved for future use.

(v) Message ID – This sixteen-bit field is used for detection of message duplication, messages of type acknowledgement/reset and confirmable.

(vi) Payload – The payload carries sensor data or resource representation.

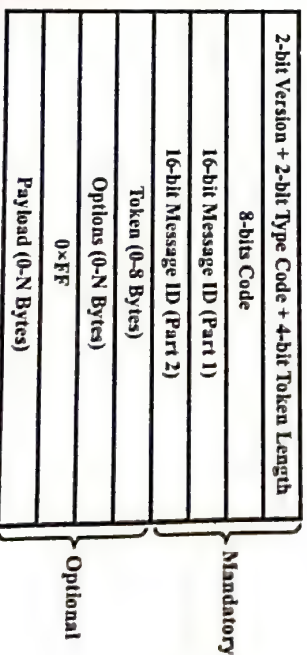| 2-bit Version + 2-bit Type Code + 4-bit Token Length	
8-bits Code	Mandatory
16-bit Message ID (Part 1)	
16-bit Message ID (Part 2)	
Token (0-8 Bytes)	
Options (0-N Bytes)	Optional
0×FF	
Payload (0-N Bytes)	

Fig. 4.11 CoAP Message Format

(vii) Options – It defines payload message type with available options are Proxy-Uri, Uri-Host, Uri-Port, Location Path, Max-Age, Uri Path Max-Age, Uri Path, Uri Query, and Token Accept.

Q.18. What types of methods are used in CoAP ? Explain.

Ans. CoAP makes use of GET, PUT, POST and DELETE methods in a same manner to HTTP and are used to manipulate the resources. Since the basic set of request methods is similar in both HTTP and CoAP, a CoAP request on HTTP resource is similar to one on a CoAP resource. A "405 method not allowed" response code should be generated in response to a unicast request with an unknown or unsupported method. As CoAP methods are similar to HTTP, they exhibit the similar properties of safe (only retrieval) and idempotent (same effect at each invoke) as HTTP. The GET method is safe while GET, PUT and DELETE methods must be performed in idempotent manner. The POST method is not idempotent because the URI embedded in the request indicates the resource that will handle the enclosed body, which can be used for data processing, a gateway to other protocols and it may create a new resource as a result of the POST. Different CoAP methods are –

(i) GET – The GET method is used to retrieve resource information identified by the request URI. In response to GET method success a 200 (OK) response is sent.

(ii) POST – The POST method creates a new subordinate resource under the parent URI requested by it to server. On successful resource creation on the server, a 201 (Created) response is sent while on failure a 200 (OK) response code is sent.

(iii) PUT – The PUT method updates or creates the resource identified by the request URI with the enclosed message body. The message body is considered as modified version of a resource if it already exists at the specified URI otherwise a new resource with that URI is created. A 200 (OK) response is received in former case whereas a 201 (Created) response is received in later case. If the resource is neither created nor modified then an error response code is sent.

(iv) DELETE – The DELETE method deletes the resource identified by the requested URI and a 200 (OK) response code is sent on successful operation.

Q.19. Give the characteristic of constrained application protocol.

Ans. Constrained application protocol has following characteristics –

(i)   Application-support layer protocol can be defined by IETF.

(ii) The access can be obtained by CoAP object or its resource as –

(a) By using a subset of response codes, which can be employed for an HTTP object

(b) By using a subset of MIME types

(c) By using the URI.

(iii) It uses request or response interaction model for communication.

(iv) With the help of ROLL network, a specialised web-transfer protocol will be employed for CoRE.

(v)   In case of resources, it employs object-model and every object can contain single or multiple instances.

(vi) In case of transmitting a request or response, an object or resource employ CoAP, DTLS and UDP protocols.

(vii) It helps the resource discovery methods and resource directory.

(viii) The URIs is employed with the help of the resource identifiers as CoAP://.......

(ix) It involves small header, message type (T), token length (TKL), ver of 4 bytes, code, token, message ID 16-bit identifier. Confirmable implies that acknowledgement will be required whereas non-confirmable implies that non-acknowledgement will be required.

(x)   On over ROLL, CoRE communication will be asynchronous.

(xi) A web client may not even be aware that it just transmitted data to an IoT device actuator resource and accessed an IoT device sensor resource. Using CoAP application cross-protocol proxies, integrates easily with the web. This can be obtained because HTTP and CoAP both share the REST model.

Q.20. Explain the CoAP Request-Response model.

Ans. CoAP resembles HTTP with request methods of GET, POST, PUT, and DELETE. They have the same properties of safe (only retrieval) and idempotent (we can invoke it multiple times with the same effects) as HTTP. These methods can be used to create, update, query and delete the resources on the server representing events of an IoT application.

In addition, CoAP defines a new method the Observe method. This method is used to implement a subscription concept build-in with the protocol. This method is not in HTTP protocol and has been constructed over HTTP in many ways. Observe method is simply a GET request with the option called observe which tell the server that this client want to get every update in this resource. The observe is define be Taken and every time a change happens to the resource the server sends a new response to the client with the same Taken.

There are three ways to send responses – Piggy-backed, separate and non-confirmable. A piggy-backed response is a response that is delivered with an acknowledgement message. Sometimes it is not possible (e.g., due to long processing time) to send the answer immediately, so first an empty acknowledgement and later the answer is sent in a separate confirmable message. Finally, if the request was received with a non-confirmable message, the response must also be sent as non-confirmable.

Response codes are similar to HTTP. For instance, 2.xx indicates success, 4.xx indicates client error and 5.xx indicates server error. Although some response codes match the HTTP status codes (eg, 4.04 and 404 "Not Found"), others have different codes (2.05 "Content" is equivalent to 200 "OK", but 2.05 is only used in response to GET) or are not represented at all (see table 4.2).

Table 4.2 CoAP Response Codes

Code	Beschreibung	Code	Beschreibung
2.01	Created	4.05	Method not allowed
2.02	Deleted	4.06	Not acceptable
2.03	Valid	4.12	Precondition failed
2.04	Changed	4.13	Request entity too large
2.05	Content	4.15	Unsupported content format
4.00	Bad request	5.00	Internal server error
4.01	Unauthorized	5.01	Not implemented
4.02	Bad option	5.02	Bad gateway
4.03	Forbidden	5.03	Service unavailable
4.04	Not found	5.04	Gateway timeout
		5.05	Proxying not supported

The request/response could happen in two ways piggy-backed or separate response. Fig. 4.12 shows the piggy-backed where the client sends the request using CON type or NON-type message and receives response ACK message immediately.
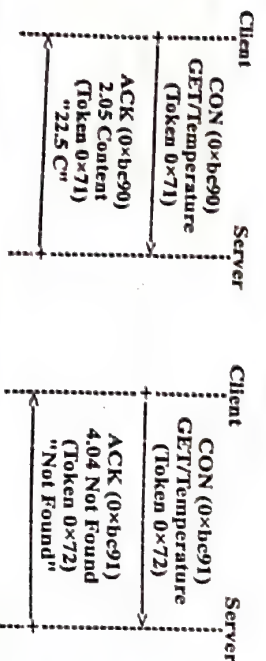

Fig. 4.12 Piggy-backed Request/Response Transmission

In the separate response message type shown in fig. 4.13 when the receiver receives a CON type message but not able to respond to this request immediately, it will send an empty ACK message immediately. When the response is ready, it will send a new CON to client and client reply a confirmable message with acknowledgement.
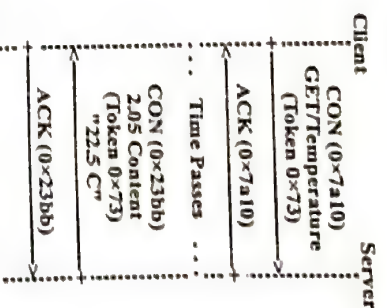
Fig. 4.13 Separate Request/Response Transmission

Q.21. What are the applications of CoAP ?

Ans. Applications of CoAP are as follows –

(i) Basic CoAP Based System Setup – The lowest level of CoAP based system consists of machines that is sensors and actuators that measures or take actions. These sensors or actuators forms a small network to interact with the outer world over CoAP. There may be need to setup a proxy which finally sends data to an HTTP server.

(ii) Real Time Condition Based Monitoring in Smart Grid – A smart grid is an intelligent power generation, distribution and monitoring system. It uses the modem information and communication technologies to gather and act on data. Today, multiple sensors enclosed in a unit can be placed on the transformer. The unit, then in turn, sends the data to the data centres by different means (PLC, GPRS, and Ethernet). They send the data over 6LowPAN with CoAP to the edge router and proxy combination. This will standardize the way manufacturers create the sensors.

(iii) Defense Equipment – Battle tank today has thousands of sensors each connected with wires. It is good to replace these tiny sensors with another tiny set of sensors based on CoAP. Intruder detection system consists of many tiny sensors to detect any intrusion. These sensors hop their data from one another to server, which connects to a secure network running standard Internet protocols. Some of these sensors bandwidth is few bytes or one byte sometimes, therefore it becomes essential that protocol selected should have low overhead. Thus making CoAP as preferred choice, moreover it also gives interface to http. However, CoAP uses duty cycling mechanism for power saving, an essential requirement for small nodes to have a long life.

(iv) Aircraft Equipment – A commercial aircraft contains miles and miles of wiring just connecting many sensors and actuators with each other, Smart CoAP based sensors and actuators may help to save weight in the aircraft, making it more efficient.

(v) Factory Instrumentation – Many manufacturing factories centrally manage and control various kinds of instruments for measuring different parameters. Sensors used in these applications are low bandwidth sensors on which standard protocols do not work because of complexity and cost. In such case a low bandwidth protocol like CoAP is effective in managing cost and providing simplicity.

Q.22. Define the terms –
(i) Server discovery (ii) Resource discovery (iii) Observe.

Ans. (i) Server Discovery – Client discovers server with the knowledge of server URL consisting of server machine address with port number at which CoAP server resides. The default port is 5683 as defined in Shelby, 2013. Therefore, the client needs to – set up a UDP connection with the server, send a GET request to the server over the given URL path (coap:// www.example.com:5683/sensor/temperature) and get a response.

(ii) Resource Discovery – CoAP uses standard methods for resource discovery. Similar to standard methods, servers have a list of available resources (with metadata about them) in the application/link-format to allow a client for discovering them and their media type. Predefined macro defines resources. Each resource needs name, path, interface, description, resource type and code. For periodic resources the actuator must know the period. A callback function performs the needed action.

(iii) Observe – Resource observation in CoAP is simply an extension of HTTP request. The CoAP GET resource checks for set condition of observe flag. The flag sets on receiving a CoAP GET resource. Client does not need to request repeatedly while the server responds by seeing the change in parameters. This enables servers to communicate the state changes as needed by the client. Both server and client have authorization to end the observation.

Q.23. Explain about message communication protocols for connected devices.

Ans. The protocols used for message communication for connected devices are as follows –

(i) CoAP-SMS – When CoAP object employs IP as well as cellular network and employs SMS, then CoAP-SMS represents a protocol. For CoAP object messages and when using cellular communication, it is an alternative to UDP-DTLS over ROLL. CoAP client or server uses SMS instead of UDP + DTLS. A CoAP client using CoAP-SMS protocol communicates to a mobile

terminal endpoint over the general packet radio service, long term evolution or high speed packet access networks.

The CoAP-SMS characteristics are given below –

(a) An URI employed like coap+sms:// in place of coap://. For instance, URI can be coap+sms://telNum/homeLocationObject/latitude for latitude of location of a home after LocationObject measures location parameters using the GPS. When transmitting the SMS to the specified telephone number (telNum), URI is employed.

(b) In case of SMS communication, a CoAP message encodes with alphabets. For 7-bit encoding of a character, an SMS message comprises of 160 characters. When an SMS-C supports 8-bit encoding, the maximum length in a CoAP message is 140 B. And an SMS-C supports 16-bit encoding and multilingual alphabets, then CoAP message is 70 B. Concatenated short messages given upto 255 B by the SMS-C supports.

(c) In cellular networks, CoAP end points have to work with subscriber identity module card for SMS. Mobile station ISDN number addresses the end points. A TP-DATA-Coding-Scheme inclusion allows the CoAP client to search short message which contains a CoAP message.

(d) Multi-casting is not supported by it.

(e) Response-to-URI-Port and Response-to-URI-Host are two additional options make originating CoAP client aware of the presence of the alternative interface CIMD and SMPP and UCP/UMI Response-to-URI-Host is a string of size ranges from 0 to 255 B. With default port number 5683, Response-to-URI-Port size is about 2 B. In case of registry, IANA registered TBD like CoAP URI option numbers.

(f) Data interchange sequences can be expressed as –

(1) MS/CoAP client transmits a SMS request (SMS-SUBMIT) to SMS-C

(2) SMS-C reports using SMS-SUBMIT-REPORT.

(3) SMS-C transmits SMS (SMS-DELIVER) to MS/CoAP Server

(4) Server reports using SMS-DELIVER-REPORT

(5) SMS-C transmits SMS-STATUS-REPORT to the client.

(g) The server provides the security by authenticating the client. During SMS data exchanges, MSISDN of the MS and SIM based security is employed.

(ii) CoAP-MQ – It represents a message queue protocol with the help of a broker and RD. CoAP endpoints behave like client and server. CoAP-MQ server provisioning in case of resource-subscription, store from the

publisher can be shown in fig. 4.14. The server also provisions in case of transmitting to the subscriber and publisher. Here, RD services represent resource discovery, directory and proxy services. The device objects communicate with the help of CoAP client and server protocols and CoAp web object with the help of DTLS like security protocol and UDP in CoAp APIs. Fig. 4.14 shows data exchanges between CoAP-MQ endpoints, CoAP-MQ clients, CoAP-MQ servers with the help of CoAP-MQ broker and CoAP-MQ broker services.
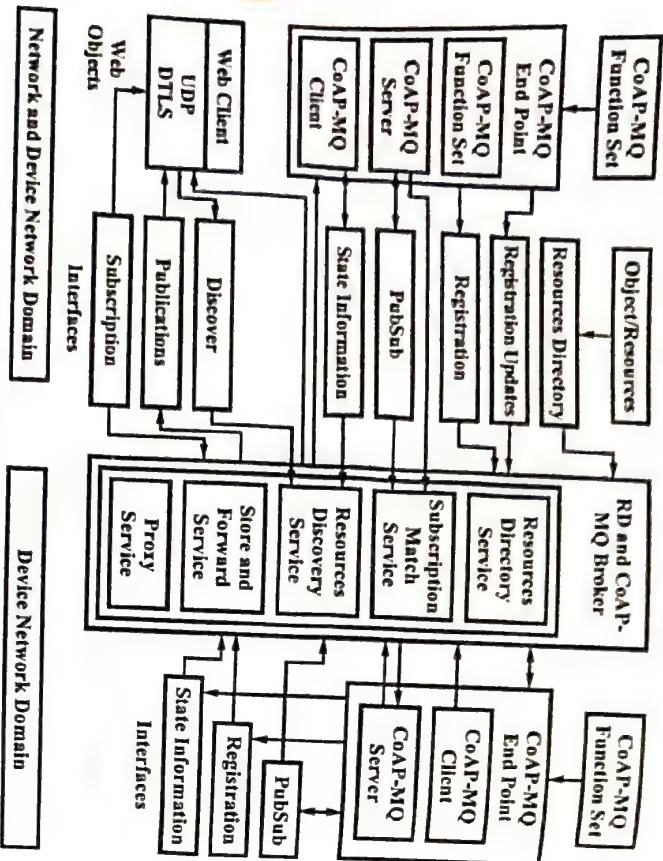

Fig. 4.14

(iii) MQTT Protocol – Refer Q.1.

XMPP, AMQP FEATURES AND COMPONENTS, AMQP FRAME TYPES

Q.24. Write short note on XMPP.

Ans. In case of messaging and presence protocols, XMPP represents an XML-dependent specification. Besides, it represents a open-source protocol recommended specification. This specification is accepted by IETF. The full form of RFC is recommendation for comments, and it represents an international

---

organization. The XMPP for CoRE is defined by RFC 6120 document. Instant messaging and presence are defined by RFC 6121 XMPP document. Address format is defined by RFC 6122 XMPP document. Message notify presence for IMs to one or many simultaneously.

Messages notify presence for the IMs to one or many simultaneously. It enables chatting between two or many users after creation of a chat room. XMPP enables IMs between many users like it employs presence-notifications and chat characteristics. XMPP activates interoperable communication for Google Talk.

Chat room represents an application software, where all those who have subscribed are given a room as view and the IMs are used among themselves.

Q.25. What are the features of XMPP ?

Ans. Following are the features of XMPP –

(i) It uses XML language.

(ii) There are three types of XMPP elements –

  (a) Message

  (b) Presence

  (c) IQ (information/query, request/response)

(iii) For open-ended stream XML elements are transmitted within the tag <stream> and corresponding end tag </stream>.

Q.26. What are the advantages and disadvantages of XMPP ?

Ans. The advantages and disadvantages of XMPP are as follows –

Advantages of XMPP –

(i) Firewall friendly

(ii) Enables pushing data, not just pulling

(iii) Provides many out-of-the-box tools for solving a wide range of problems.

(iv) Strong authentication and security.

Disadvantages of XMPP –

(i) More overhead than HTTP for simple chores

(ii) Specialized implementations still needed

(iii) Stateful protocol

(iv) Community and deployments are not as broad as HTTP.

Q.27. Write short note on AMQP.

Ans. Advanced message queuing protocol (AMQP) is another OASIS standard that was designed for the financial industry, runs over TCP, and uses publish/subscribe architecture, similar to MQTT. The main difference in these standards is that the broker is divided into two main components – exchange

and queues. The exchange component is responsible for receiving publisher messages and distributing them to queues following predetermined roles. Subscribers connect to those queues, which basically represent the topics, and get the sensory data whenever they are available.

Q.28. Enlist the features of AMQP.

Ans. The features of AMQP are as follows –

(i) It is a secure protocol, it uses TLS/SSL protocols over TCP.

(ii) It ensures reliable communication by providing three message delivery guarantees (at most once, at least once, and exactly once).

(iii) It can send a huge amount of messages per second. It has been found that the AMQP can process 300 million messages per day in a distributed environment of 2000 users.

(iv) It provides an asynchronous publish/subscribe communication.

(v) It offers store-and-forward feature to ensure reliability even after network disruptions.

Q.29. Explain the architecture of AMQP.

Ans. AMQP is an open customary application layer protocol for middleware messaging protocol. It is a uniform protocol by OASIS. Currently, it is broadly used in business and commercial platforms. It is a unique protocol because it supports point to point and publisher/subscriber models, routing and switching. Moreover, it has the ability to do message orientation, queuing, switching reliability and security.

AMQP is different from MQTT, and has more advantages. It stores the data, and then forwards it. This feature works when the network is troubleshooting that time guarantees reliability. AMQP conforms reliability with the following message-delivery guarantees –

(i) *At Most Once* – The message is sent only once, whether received or not.

(ii) *At Least Once* – The message is sent on time.

(iii) *Exactly Once* – The message is sent only once.

In AMQP, the security held with the TLS protocols are above TCP. Many studies have pointed out that AMQP has low bandwidths and achievement rate. According to one study, AMQP can direct an enormous number of messages per second. Furthermore, AMQP environment with 2000 users from five continents is able to process 300 million messages per day. Fig. 4.15 shows that AMQP component, the broker is divided into two parts of exchange and the queue that hold the communication. The exchange receives publishers' messages and gives them to the queue. The queues send the messages and the data to subscribers.
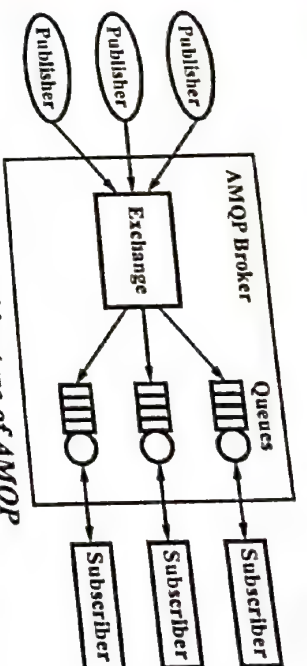
Fig. 4.15 Architecture of AMQP

AMQP is widely run in IoT devices which concentrate on message exchange and communication. It supports many languages, and decreases communication problems between dissimilar devices. AMQP is a highly flexible communication protocol that can work in different platforms and environment applications. It supports industrial environment applications, yet it is unsuitable for constrained environments and automation discovery mechanisms.

Q.30. Discuss the following –

(i) *Delimiting frames*   (ii) *Method frames*

(iii) *Content frames*   (iv) *Heartbeat frames.*

Ans. (i) *Delimiting Frames* – TCP/IP is a stream protocol, i.e. there is no in-built mechanism for delimiting frames. Existing protocols solve this in several different ways –

(a) Sending a single frame per connection. This is simple but slow.

(b) Adding frame delimiters to the stream. This is simple but slow to parse.

(c) Counting the size of frames and sending the size in front of each frame. This is simple and fast, and usually preferred.

(ii) *Method Frames* – These frames carry the high-level protocol commands (called as "methods"). One method frame carries one command. The method frame payload has this format –

```
  0      2        4
+------+---------+-------------+
|Class-id|Method-id|Arguments ...|
+------+---------+-------------+
 Short    Short        ...
```
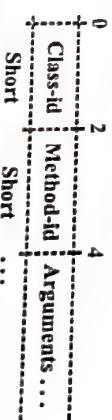
A method frame is processed in following steps –

(a) Read the method frame payload.

(b) Unpack it into a structure. A given method always has the same structure, so we can unpack the method rapidly.

(c) Check that the method is allowed in the current context.

(d) Check that the method arguments are valid.

(e) Execute the method.

Method frame bodies are constructed as a list of AMQP data fields (bits, integers, strings and string tables). The marshalling code is trivially generated directly from the protocol specifications, and can be very rapid.

(iii) Content Frames – Content is the application data we carry from client-to-client via the AMQP server. Content is, roughly speaking, a set of properties plus a binary data part. The set of allowed properties are defined by the basic class, and these form the "content header frame". The data can be any size, and may be broken into several (or many) chunks, each forming a "content body frame".

Looking at the frames for a specific channel, as they pass on the wire, we might see something like this –

```
[method]
[method] [header] [body] [body
[method]
…
```

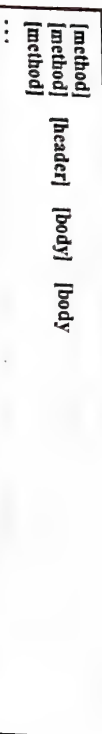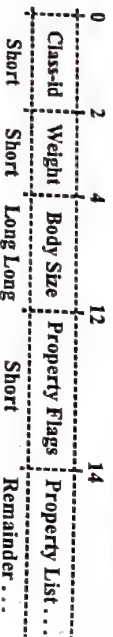Certain methods (such as Basic.Publish, Basic.Deliver, etc.) are formally defined as carrying content. When a peer sends such a method frame, it always follows it with a content header and zero or more content body frames.

A content header frame has this format –

0	2	4	12	14
Class-id	Weight	Body Size	Property Flags	Property List...
Short	Short	Long Long	Short	Remainder...

We place content body in distinct frames (rather than including it in the method) so that AMQP may support "zero copy" techniques in which content is never marshalled or encoded. We place the content properties in their own frame so that recipients can selectively discard contents they do not want to process.

(iv) Heartbeat Frames – Heartbeating is a technique designed to undo one of TCP/IP's features, namely its ability to recover from a broken physical connection by closing only after a quite long time-out. In some scenarios we need to know very rapidly if a peer is disconnected or not responding for other reasons (e.g. it is looping). Since heartbeating can be done at a low level, we implement this as a special type of frame that peers exchange at the transport level, rather than as a class method.

●●

UNIT 5

IoT PLATFORMS, ARDUINO, RASPBERRY PI BOARD, OTHER IoT PLATFORMS

Q.1. What is an IoT platform ?

Ans. When we are developing some application, platform is one which allows us to deploy and run our application. A platform could be a hardware plus software suite upon which other applications can operate. Platform could comprise hardware above which operating system can reside. This operating system will allow application to work above it by providing necessary execution environment to it.

IoT platforms (more specifically IoT application platforms) provide a comprehensive set of generic, i.e. application independent functionalities which can be used to build IoT applications. When there is only one communication link between devices of one type with another device of same type then, a system of specific service can be set up. But in case of communication among devices of multiple types, there is a need of some common standard application platform which hides heterogeneity of various devices by providing a common working environment to them. An IoT application platform is a virtual solution, means it resides over cloud. Data is the entity that drives business intelligence and every device has something to talk with other device that is data. By means of cloud connectivity, IoT application platform translates such devices data into useful information. So it provides user a means to implement business use cases and enables predictive maintenance, pay-per-use, analytics and real time data management. Thus, IoT application platforms provide a complete suite for application development to its deployment and maintenance.

Various IoT platforms are now-a-days available that can be used for developing an IoT solution. Thingworx, Thingspeak and Grovestreams are examples of IoT platforms.

Q.2. Write in brief about ThingWorx platform.

Ans. ThingWorx is the first software platform designed to build and run the connected world applications. ThingWorx focuses on reducing the time,
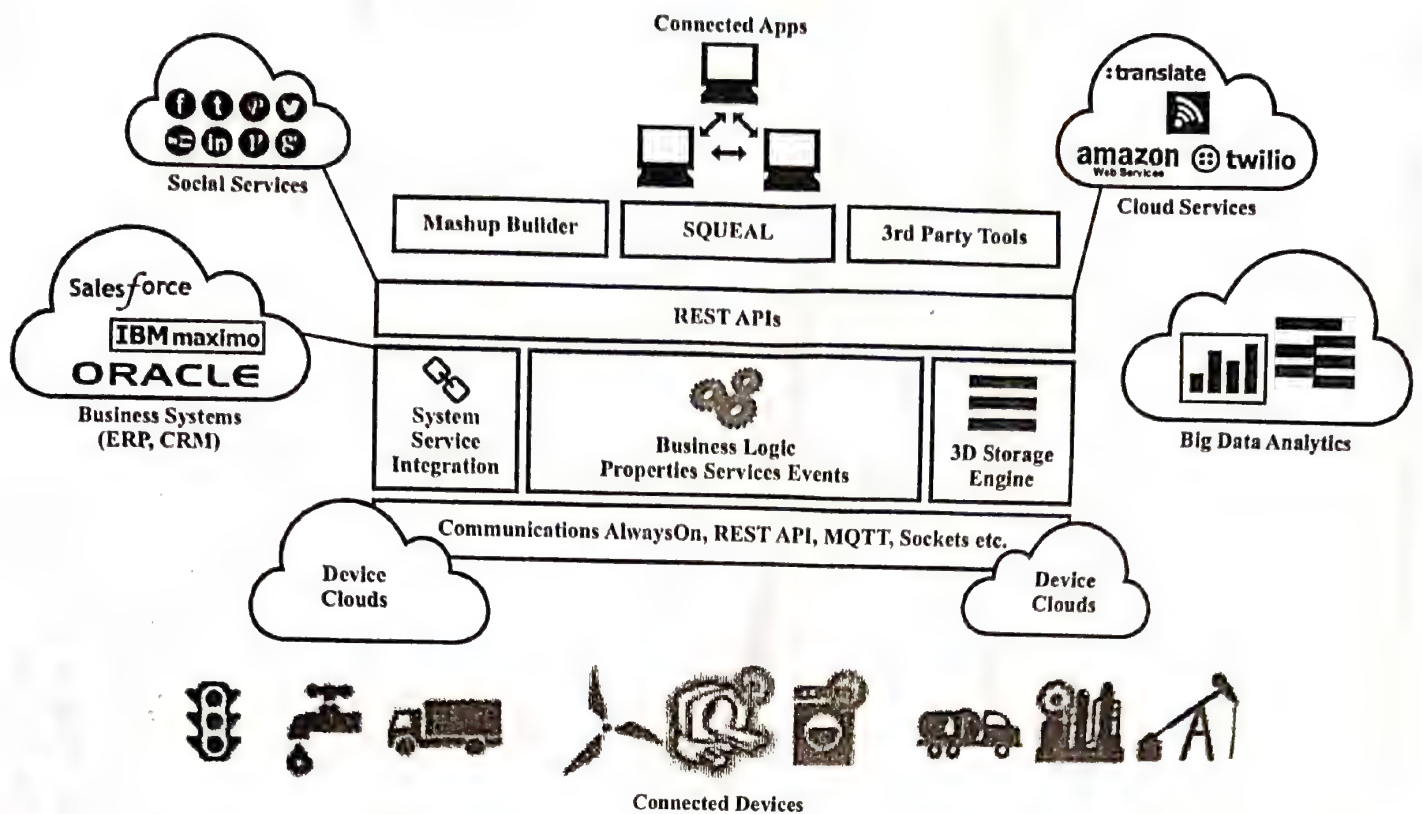
Fig. 5.1 ThingWorx IoT Foundation

cost, and risk required to build innovative Machine-to-Machine (M2M) and Internet of Things (IoT) applications. ThingWorx accelerates IoT application development by compressing the design-develop-deploy cycle and reducing time to market. ThingWorx allows us deploy exactly the way we want either from cloud, on premise, federated or embedded, to fit the needs of any scenario. ThingWorx supports rapid creation of smart applications like – Smart Agriculture, Smart Cities, Smart Grid, Smart Water, Smart Buildings, and Telematics. It is a development suite that enables complete application design, runtime, and intelligence environment. The ThingWorx Internet of Things foundation is powered by a ThingWorx's following leading products and services – ThingWorx Composer, Codeless Mashup Builder, Event-Driven Execution and 3D storage, ThingWorx SQUEAL, ThingWorx Edge MicroServer.

ThingWorx IoT platform provides device cloud to connect millions of devices with IoT application. It provides always on communication using REST, MQTT and sockets. Above the layer of communication there are system service integration, 3D storage engine and business logic. System service integration interact with business systems like ERP, CRM etc. 3D storage engine enables big data analytics. Above it there is a layer for REST APIs that helps to implement and use social services and cloud services. Then the data is presented via various visualization techniques.

Q.3. What are the characteristics of ThingWorx IoT platform ?

Ans. The characteristics of ThingWorx IoT platform are as follows –

(i) Flexible Connectivity Options – ThingWorx "inclusive" connectivity strategy maximizes market opportunity and minimizes integration efforts.

(ii) ThingWorx SQUEAL TM (Search, Query & Analysis) – With the interactive search capabilities of ThingWorx SQUEAL, users can now correlate data that delivers answers to key business questions. Pertinent and related collaboration data, line-of-business system records, and equipment data get returned in a single search, speeding problem resolution and enabling innovation.

(iii) Model-Based Development – Reduces development time by ten times.

(iv) Codeless Mashup Builder – Facilitates rapid creation of IoT applications.

(v) ThingWorx Composer – It makes it easy to model the things, business logic, visualization, data storage, collaboration, and security required for a connected application.

Q.4. Discuss about the ThingSpeak IoT platform.

Ans. ThingSpeak is an application platform for the development of IoT systems. It can help us to build the application which works upon the data

collected by sensors. ThingSpeak is an open data platform for IoT application development. ThingSpeak is the perfect complement to an existing enterprise system to tap into the Internet of Things. It provides the ability to integrate our data with a variety of third-party platforms, systems and technologies, including other leading IoT platforms such as ioBridge and Arduino. ThingSpeak channel is used to send and store data. Each channel has, eight fields that can hold any type of data, three location fields, and one status field. After creating a ThingSpeak channel, one can publish data to the channel, the data can be processed and application can retrieve the data.

ThingSpeak platform provides following functionality to support IoT system –

(i) Collect – Sends sensor and device data collected from it to the cloud so that the data can be further analysed.

(ii) Analyse – ThingSpeak can analyse the data received from sensors of devices and can derive the virtual representation of the data.

(iii) Act – Based upon the analysis, it will trigger the action to enable functioning of IoT system and application.

Fig. 5.2 shows ThingSpeak IoT with Electric Imp platform. Here Electric Imp is a platform which provides connectivity of Wi-Fi devices with cloud services. It provides access to the Electric Imp modules. Once the connectivity to such modules is simplified by Electric Imp cloud, data services from ThingSpeak are used to get IoT experience.
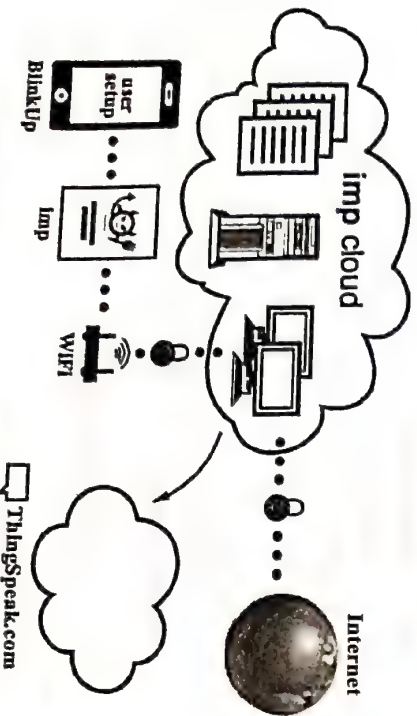

Fig. 5.2 Electric Imp and ThingSpeak IoT

Q.5. What are the features of ThingSpeak IoT platform ?
Ans. The key features of ThingSpeak IoT platform are as follows –
(i) It provides real time collection of data storage

(ii) Data analytics and visualization using MATLAB
(iii) Device communication
(iv) Open API support
(v) Provides geolocation data
(vi) Facilitates plugins.

Q.6. Discuss about the GroveStreams IoT platform and its advantages.

Ans. GroveStreams is one of the most powerful platforms in the cloud providing near real-time decision making capabilities to millions of users and devices. GroveStreams is a young, but clear-headed IoT analytics vendor whose technology was designed to solve the ease-of-deployment problem from the get-go. The aim of GroveStreams is to minimize deployment times by having an integrated, end-to-end, IoT-specific analytics platform that handles data ingestion, analytic calculation definition and analytics visualization and doesnot require writing script or code. GroveStreams is also designed to handle the volume and breadth of streaming data that is common within IoT systems. In particular, it can aggregate hundreds or thousands of event streams and then display them in a few graphical metrics to provide top-level views of complex system operation. The platform can store up to 80 million data points per raw input stream, so longitudinal analytics have a lot of headroom. Sample granularity can accommodate five samples per second. JSON encapsulation is supported, with each RESTful PUT API call capable of up to 250 MB in the message body, so good flexibility to consume a wide range of IoT device data.

Because the platform is cloud-deployed, they are continuously enhancing the product. Their most recent addition allows authorization to be centrally managed and applied for user groups and communities. Hierarchical authorization rights can be applied to mirror responsibilities in a hierarchically managed organization. Some of the advantages of this platform are as follows–

(i) Simple to put data in and take data out of the store.
(ii) Built with BigData technology. Scales to nearly unlimited size.
(iii) A user can get started with a free account.Pricing is data driven.
(iv) Location/Map integration is also available.
(v) The roll-up stream data is a really efficient way to get to the data needed.
(vi) It is possible to run very complex expressions on any stream in the store.
(vii) The dashboards to visualize the data are really good and easy to use.
(viii) Users can get alerts upon arrival for alarm type streams.
(ix) Blueprinting an organization makes setting up new organizations really easy.

Q.7. Explain in brief about Arduino with IoT applications.

Ans. Famous AVR MCU-based products are Arduino boards, modules and shields. On the connections pins, in-circuit connections and sockets, each board includes clear markings. Hence, simplify the prototyping of embedded platforms for IoTs and each Arduino board is simple to work for do-it-yourself (DIY). Hence, Arduino boards are simple to program because of IDEs are open source. In the beginning with electronics and programming, Arduino Uno board is a mostly used board. Now-a-days, entire Arduino family Uno is widely used. The board has analog input pins and PWM pins. These pins are used to connect sensors, actuators and analog circuits. On-off states, set of on-off states, digital inputs from sensors, digital outputs to actuators and other digital circuits can connect with the help of the board's digital input output pins. Wireless connection to a ZigBee, Bluetooth LE, WiFi, GSM, or RF module or a wired connection to Ethernet LAN for the internet can be established by a board with a shield inserted into it. Arduino Ethernet, Arduino WiFi and Arduino GSM shields are development boards for IoT devices. Arduino Gemma, LilyPad, LilyPad Simple/SimpleSnap and LilyPad USB are development boards for the wearable devices.

In the MCU, a board has pre-programmed bootloader. On to the AVR MCU chip, bootloader software program embeds. Using USB connection to a computer, bootloader can also be downloaded. Use of the AVR platform with the Arduino IDE is allowed by bootloader. The board function is allowed by bootloader. By default after bootloading the board does not require operating system. When necessary, an operating system can be embedded. The program are pushed into the MCU using USB port of the board, if a developer develops the codes using an IDE. After developing-testing-debugging cycle, codes are pushed. Using the editor in an integrated development environment (IDE), a developer writes the codes. After coding, these are downloaded onto the board, tested and debugged. A USB port interconnecting the board and external computer with an IDE is used for downloading. Architecture of Arduino Fio board with ethernet shield is shown in fig. 5.3.
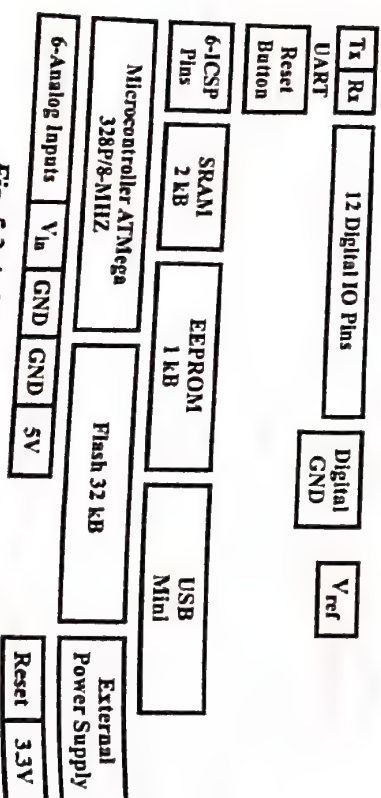

Tx	Rx		
UART		12 Digital IO Pins	
Reset Button		Digital GND	Vref
6-ICSP Pins	SRAM 2 kB	EEPROM 1 kB	USB Mini
	Microcontroller ATMega 328P/8-MHZ	Flash 32 kB	
6-Analog Inputs	Vin	GND	GND	5V
			External Power Supply	Reset	3.3V

Fig. 5.3 Arduino Fio Board Architecture

Applications – An embedded-device data connectivity to the internet and data store on cloud can be an Arduino application. Internet of streetlights is an example of an Arduino application for IoT. Applications where the device does not need intensive computing and graphics is obtained by Arduino board. Light-emitting diodes, wearable devices, health monitoring or fitness devices, watches, sensors and actuators connected smartly through the internet are applications which using things. The developments element are open sources which use the computer with Windows, Arduino Linux distribution or a MAC.

Q.8. Give the classifications of Arduino features of UNO and other IoT device and wearable device boards.

Ans. Table 5.1 shows the features of Arduino UNO and IoT and wearable device boards.

Table 5.1

Board/ Shield	WiFi/ GSM/ Ethernet	USB/ UART	n-bit PWM/ Digital IOs/ Analog in/out	Flash/SRAM/ EEPROM	Operating/ Input V	AVR Micro-controller/ Clock	Application
DUE	0/0/0	2 micro/4	12/54/2/12	512 MB/96 kB/0 kB	3.3V/7V-12V	ATSAMS X 81	Fast computations, ARM based MCU are application
UNO	0/0/0	Standard 1	6/14/6/0	32 kB/2 kB/1 kB	5V/7V-12V	ATmega 328/16 MHz	Getting started with electronics and coding
Lilypad USB	0/0/0	Micro/0	4/9/4/0	32 kB/2.5 kB/1 kB	33V/3.8V-5V	ATmega 32U4/8 MHz	Wearable
LilyPad Simple Snap	0/0/0	0/0	4/9/4/0	8 kB/512B/512 B	2.7-5.5V/2.5-5.5V	ATmega 328P/8 MHz	Wearable
LilyPad	0/0/0	0/0	6/14/6/0	16 kB/1kB/512 B	2.7-5.5V/2.7-5.5V	ATmega 168V/8 MHz ATmega 328P/8 MHz	Wearable
Gemma	0/0/0	Micro/0	2/3/1/0	8 kB/512B/512 B	3.3 V/4-16V	ATtiny 85/8 MHz	Wearable
Fio	0/0/0	Mini/1	6/14/8/0	32kB/2kB/1 kB	3.3 V/3.7V-7V	ATmega 328P/8 MHz	IOT
Yun	0/0/0	Micro/1	0/12/20/7	(i) 32kB/2.5MB/1 kB (ii) 64kB/16MB/1 MB	5V	(i)ATmega 32U4/16 MHz (ii) AR9331Linux 400 MHz	IOT
Ethernet	0/0/1	Standard/0	4/14/6/0	32kB/2kB/1kB	5V/7V-12V	ATmega 328 16 MHz	IOT

Q.9. Explain Arduino boards. List the feature which make Arduino boards widely used.

(R.GP.V., May 2018)

Ans. Arduino Board – Refer Q.7.

Features – There are several features which make Arduino boards widely used –

(i) During the editing-testing-debugging cycle, multiple coding of the board and for development of number of new prototype using the same.

(ii) Prototyping ease.

(iii) On the board, flexibility and ease of assembling modules.

(iv) Software runs on multiple OS like Linux, Windows, Mac OS-X.

(v) Operating system and IDE latest version are open source.

(vi) Hardware open source and extensible using the modules, shields and other circuits with open version of IDE, software modules and codes from other designers.

(vii) Extensible source code, schematics, software, middleware and IDE are open C commands and statements for using AVR ports, serial interfaces and other functional units of the MCU mean AVR-C. The AVR-C codes extend on coding in C++ and the libraries can be added with additional programs.

Q.10. How to connect Arduino USB to internet, Arduino to Internet and Arduino to WiFi ?

Ans. Connecting Arduino USB to Internet – USB is supported by Arduino board IDE. Using IDE USB port functions, USB port connects to a mobile with an Internet connection. After that using network interface cards, a computer connects to the Internet.

Connecting Arduino to Internet – Ethernet protocol library is supported by Arduino IDE. Library is defined by set of codes that allows use of functions of the library for particular purposes. Ethernet LAN directly connects to the network router. The ethernet client mode of shield is employed, if the shield transmits data of the device to the cloud. The ethernet server mode is employed if the shield transmits data of the device to the computer.

Connecting Arduino to WiFi – WiShield library is supported by Arduino IDE. WiShield connects to a network router wirelessly. However from the power supply adequate energy for WiFi communications is necessary.

Example – The header files in the WiShield library are –

Need to use WiShield is that the program predefines –

(i) Type of network, infrastructure fixed network or ad-hoc network

```
#include<util.h>
#include<SPI.h>
#include<WiServer.h>
```

(ii) Types of security – 1 WEP
2 WPA
3 WPA2

(iii) WEP key – 128-bit when security type = 1

(iv) SSID – A unique ID of 32 characters and is used for naming wireless networks

A set of networked interconnected devices which communicate using SSID of the set is known as service set.

Q.11. Write a short note on bootloader.

Ans. A system software that loads or embeds into a microcontroller chip or computing platform to let the system being its function is known as bootloader. In a device, the bootloader firmware contains at flash of microcontroller. It allows communication with a computer which include IDE. Generally, the IDE has APIs, compilers, libraries, RTOs, simulator, assembler, editor, debugger, emulators, code burner, logic analyser, and other software for integrated development of the system. An IDE can be open source. Development of code is allowed by IDE on a computer. After that downloading of codes on to Arduino board. Into flash memory, the code is stored by burner. Hence, into the device, the special application codes are embedded.

Q.12. Write short note on programming embedded device Arduino platform using IDE.

Ans. By using avr-gcc tools, Arduino board can be programmed. The Arduino board includes a pre-installed bootloader that embedded into the firmware. By using a graphical cross-platform IDE, Arduino programmer develops the codes. Simplicity is given by Arduino, based on processing language and makes the programming easy. IDE of Arduino board also has simplicity. A computer that runs the IDE is connected with boards. This loader running is enabled by bootloader program and handover the control by bootloader program. By the usage of interrupt handing functions for each task, the Arduino bootloader provisions for multitasking. Assigning multiple values of a number n for the tasks is responsible for multitasking. The interrupt handing function n is called for execution if an instruction for interrupt. For a special device platform, the IDE has a set of software modules, which provide the software and hardware environment for developing and prototyping the software. As per the computer operating system, a computer downloads an suitable IDE version. Generally Windows, Mac OS X or Linux runs on computer. To push the developed codes into a board the bootloader permits the computer using the Arduino IDE through a USB cable or a labelled serial port. As done in a computer where bootloader loads the operating system from the secondary disk like hard disk, Arduino bootloader need not initiate the upload of OS.

Q.13. How to develop a code through Arduino IDE?

Ans. For the code, Arduino IDE functions like a file editor, using the processing environment and library functions. Automatic indentation, highlights the syntax of the codes, and matches braces are provided by the editor. The edited file first compiles and checks and find error, if there is no error than allows pushing of codes for embedding onto the board via USB port. Arduino is simple because it requires only two functions setup( ) and loop( ) to define executable program functions for the board. The function runs at the begin and is used for initialising settings is setup( ) function. A program in endless loop using statement while (true) {statements;} which runs until the power off is loop ( ) function.

Q.14. Explain Raspberry Pi with its components parts.

Ans. A low-cost mini-computer is known as Raspberry Pi. Its physical size is same as a credit card. Raspberry Pi runs different version of Linux. It can perform almost all task which is performed by a normal desktop computer. Raspberry Pi also permits interfacing sensors and actuators through the general purpose I/O pins. Raspberry Pi runs Linux operating systems and also supports Python out of the box. The Raspberry Pi board with its different components is explained below and shown in fig. 5.4.
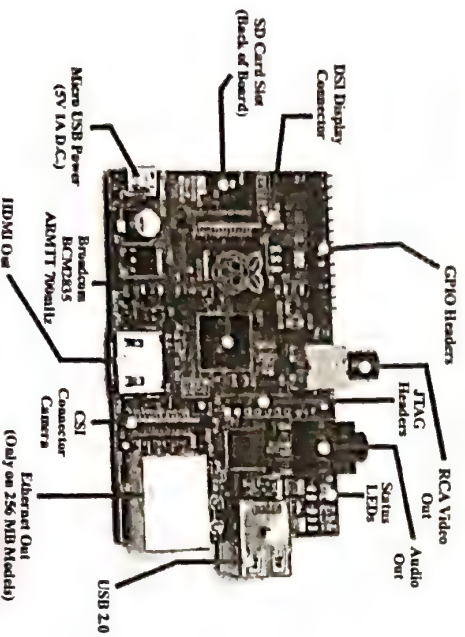

Fig. 5.4 Raspberry Pi Board

(i) Processor and RAM – Raspberry Pi uses ARM processor. The new version of Raspberry Pi supports 700 MHz Low Power, ARM1176JZ-F processor and 512 MB SDRAM.

(ii) SD Card Slot – In Raspberry Pi, operating system and storage are not in-built. So that in this case, we can plug-in SD card loaded with Linux image to the SD card slot. For installing this software on Raspberry Pi, we need atleast 8 GB SD card.

USB connector for power input which is used by power supply. In Raspberry Pi, USB ports give current upto 100 mA. An external USB powered hub is needed for connecting devices which draw current more than 100 mA.

(iv) USB Port – Raspberry Pi has two USB2.0 ports.

(v) Ethernet Ports – Raspberry Pi has in-built standard RJ45 ethernet port. To give Internet connectivity in Raspberry Pi, we can use Ethernet cable or USB WiFi adapter.

(vi) HDMI Output – Raspberry Pi has in-built HDMI port. HDMI port gives both audio and video output. By using HDMI cable can connect Raspberry Pi to monitor. For the monitors which do not have HDMI ports but only have DVI port, in that case we can use HDMI to DVI cable or adapter.

(vii) Composite Video Output – Raspberry Pi has in-built RCA jack for composite video output. This RCA jack supports both NTSC and PAL video output. It is used for connecting old television which have RCA input only.

(viii) Audio Output – Raspberry Pi has in-built 3.5 mm audio output jack. Audio jack is used for giving audio output to old television along with the RCA jack for video. In comparison to HDMI output, the audio quality from audio jack is not good.

(ix) GPIO Pins – Raspberry Pi is in-built with multiple general purpose input/output pins. Raspberry Pi general purpose input/output pins (GPIO) headers are shown in fig. 5.5. Raspberry Pi includes with four types of pins – I2C interfacing pins, serial Rx and Tx pins, true GPIO pins and true GPIO pins.
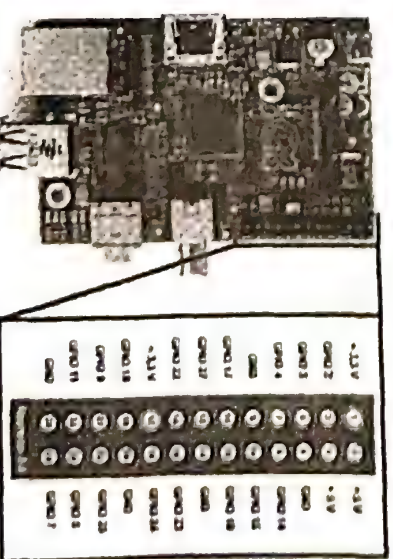
Fig. 5.5 Raspberry Pi GPIO Headers

(x) Display Serial Interface (DSI) – Raspberry Pi has in-built DSI connector which is used for connecting an LCD panel to Raspberry Pi.

(xi) Camera Serial Interface (CSI) – Raspberry Pi has in-built CSI connector which is used for connecting a camera module to Raspberry Pi.

(xii) *Status LEDs* – Raspberry Pi has five in-built status LEDs. These Raspberry Pi status LEDs with their functions are given in table 5.2.

Table 5.2

Status LED	Function
100	100 Mbit LAN attached
LNK	Activity of Link/Network
FDX	Full duplex LAN attached
PWR	3.3V power is available
ACT	SD card access

Q.15. Explain Linux on Raspberry Pi.

Ans. Raspberry Pi has multiple versions of Linux which are as follows –

(i) *Raspbian* – Raspbian Linux is a Debian Wheezy port which is made for Raspberry Pi. This is the suggested Linux for Raspberry Pi.

(ii) *RISC OS* – It is a very fast and compact operating system.

(iii) *OpenELEC* – It is a fast and user-friendly XBMC media-center distribution.

(iv) *RaspBMC* – It is an XBMC media-center distribution for Raspberry Pi.

(v) *Pidora* – It is a Fedora Linux modified for Raspberry Pi.

(vi) *Arch* – It is an Arch Linux port for AMD devices.

Raspbian Linux desktop on Raspberry Pi is shown in fig. 5.6.
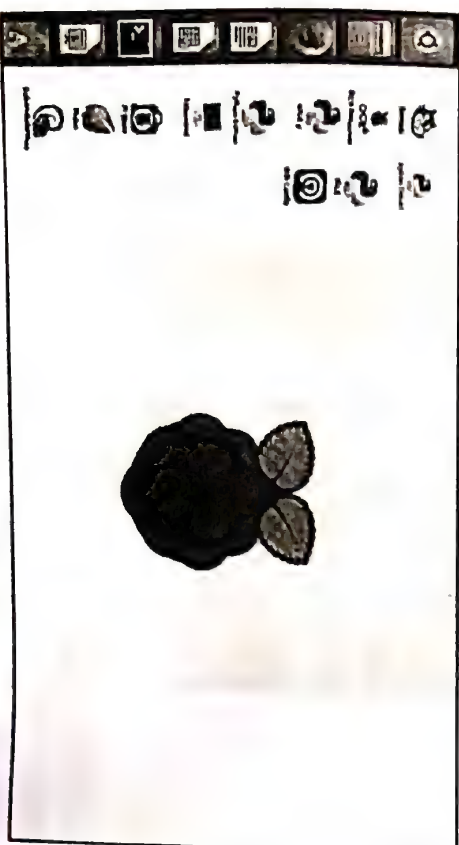
Fig. 5.6 Raspbian Linux Desktop

Default file explorer on Raspbian is shown in fig. 5.7.
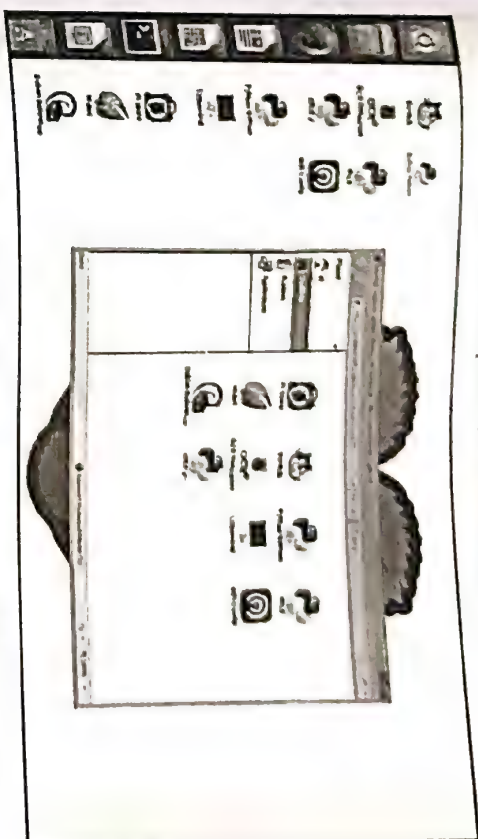
Fig. 5.7 File Explorer on Raspberry Pi

Default console on Raspbian is shown in fig. 5.8.
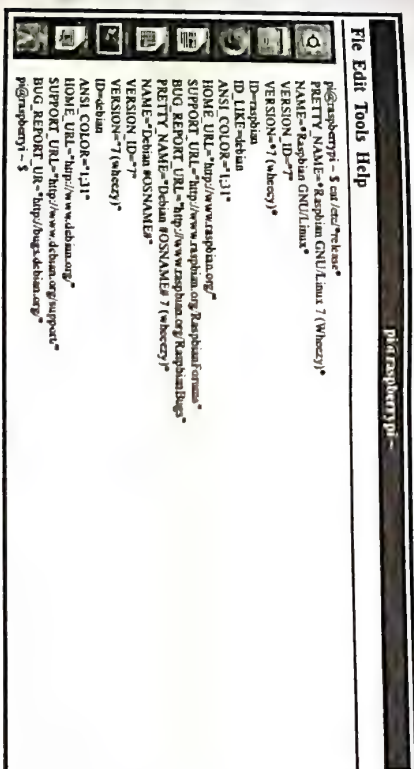
Fig. 5.8 Console on Raspberry Pi

Default browser on Raspbian is shown in fig. 5.9.

To configure Raspberry Pi, the raspi-config tool is used which begins from the command line ($raspi-config) as shown in fig. 5.10. We can expand root partition with the help of configuration tool to fill SD card, change password, change memory split, set keyboard layout, set locale and time zone, change boot behavior and enable or disable SSH server. To use the whole space on the SD card we should expand the root-file system.

Fig. 5.9 Browser on Raspberry Pi


```
┌─ Raspberry Pi Software Configuration Tool (raspi-config) ─┐
Setup Options
  1  Expand Filesystem              Ensures that all of the SD card s
  2  Change User Password           Change password for the default u
  3  Enable Boot to Desktop/Scratch Choose whether to boot into a des
  4  Internationalisation Options   Set up language and regional set
  5  Enable Camera                  Enable this Pi to work with the R
  6  Add to Rast rack               Add this Pi to the online Raspber
  7  Overclock                      Configure overclocking for your P
  8  Advanced Options               Configure advanced settings
  9  About raspi-config             Information about this configurat

            <Select>          <Finish>
```

Fig. 5.10 Raspberry Pi Configuration Tool

Raspberry Pi has an in-built HDMI output, however it is more convenient to access the device with a SSH or VNC connection. This eliminates the need for a separate display for Raspberry Pi and we can use Raspberry Pi from our laptop or desktop computer. The frequently used commands on Raspberry Pi are given in table 5.3.

Table 5.3

S.No.	Command	Function	Example
(i)	cd	Change directory	cd /home/pi
(ii)	mkdir	Make directory	mkdir /home/pi/new
(iii)	ls	List files and folders	ls /home/pi
(iv)	lsusb	List USB devices	lsusb
(v)	cat	Show file contents	cat file.txt

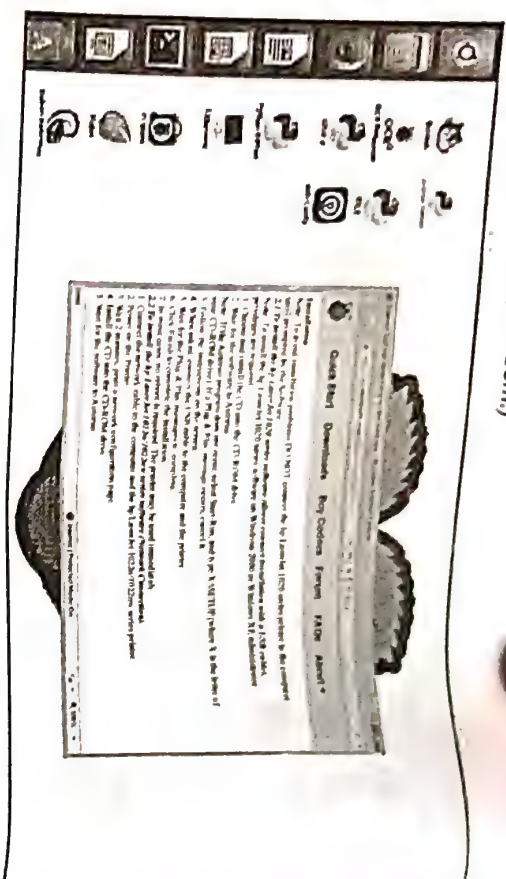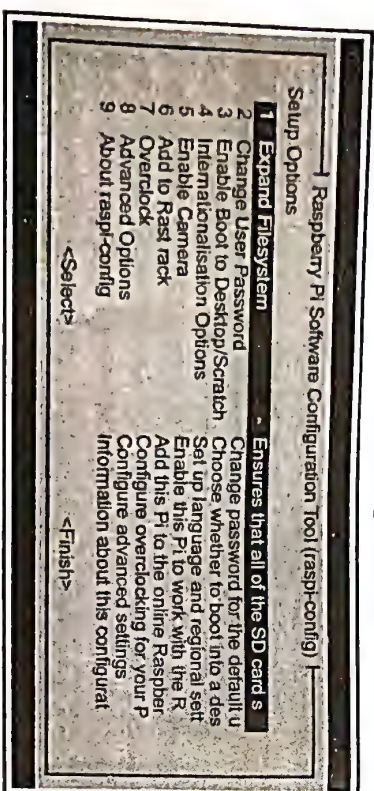|			
(vi)	locate	Search for a file	locate file.txt
(vii)	pwd	Print name of present working directory	pwd
(viii)	mv	Move (rename) file	mv sourceFile.txt destinationFile.txt
(ix)	rm	Remove file	rm file.txt
(x)	reboot	Reboot device	sudo reboot
(xi)	wget	Non-interactive network downloader	wget http://example. com/file.tar.gz
(xii)	tar	Extract/create archive	tar-xzf foo.tar.gz
(xiii)	netstat	Print network connections, routing tables, interface statistics	netstat-lntp
(xiv)	ifconfig	Configure a network interface	ifconfig
(xv)	df	Report file system disk space usage	df-Th
(xvi)	grep	Print lines matching a pattern	grep-r "pi"/home/
(xvii)	shutdown	Shutdown device	sudo shutdown-h now

Q.16. Discuss Raspberry Pi interface.

Ans. Serial, SPI, I2C interfaces are used by Raspberry Pi for data transfer –

(i) Serial – In Raspberry Pi, serial interface has receive (Rx) and transmit (Tx) pins for communication with serial peripherals.

(ii) SPI – SPI stands for Serial Peripheral Interface, which is a synchronous serial data protocol used for communicating with one or more peripheral devices. For the connection of SPI, there is one master device and one or more peripheral devices. For SPI interface, five pins are available on Raspberry Pi as given below –

(a) CE0 (Chip Enable 0) – To enable or disable electronic device.

(b) CE1 (Chip Enable 1) – To enable or disable electronic device.

(c) SCK (Serial Clock) – Clock created by master to synchronize data transmission.

(d) MOSI (Master Out Slave In) – Slave line for transmitting data to the master.

(e) MISO (Master In Slave Out) – Master line for transmitting data to the peripherals.

(iii) I2C – I2C interface enables synchronous data transfer with two pins like SDA (data line) and SCL (clock line). On Raspberry Pi, the I2C interface pins enable you to connect hardware modules.

Q.17. Explain in brief about programming Raspberry Pi with Python.

Ans. Raspberry Pi runs Linux and supports Python out of box. Means we can run any Python program which runs on a general computer. On Raspberry Pi, it is the general purpose input/output capability given by the GPIO that makes it useful device for IoT. Using the GPIO pins and the SPI, I2C and serial interfaces, we can interface a wide variety of sensor and actuators with Raspberry Pi. For instance, sending data to a server, sending an email, triggering a relay switch, input from the sensors connected to Raspberry Pi can be processed and multiple actions can be taken.

Q.18. How to control a LED with Raspberry Pi.

Ans. Switching LED on/off from Raspberry Pi Console –

```
$echo 28 > /sys /class /gpio/ export
$cd /sys /class /gpio/ gpio28
#Set pin 28 direction to out
$echo out > direction
#Turn LED on
$echo 1 > value
#Turn LED off
$echo 0 > value
```

For Blinking LED, Code in Python –

```
import RPi.GPIO as GPIO
import time
GPIO.setmode (GPIO.BCM)
GPIO.setup (28, GPIO.OUT)
While True:
GPIO.output (28, True)
time.sleep (2)
GPIO.output (28, False)
time.sleep (2)
```

Fig. 5.11 represents the schematic diagram of connecting an LED to Raspberry Pi. The program switching LED on/off indicates how to turn the LED on/off from command line. In this case, the LED connects with GPIO pin 28 then we can connect the LED with any other GPIO pin. According to the coding of blinking LED code, LED will be blinked on every two second. The code uses the RPi. On Raspberry Pi, GPIO module to control the GPIO. In this code we set pin 28 direction to output and then write True/False alternatively after a delay of two seconds.
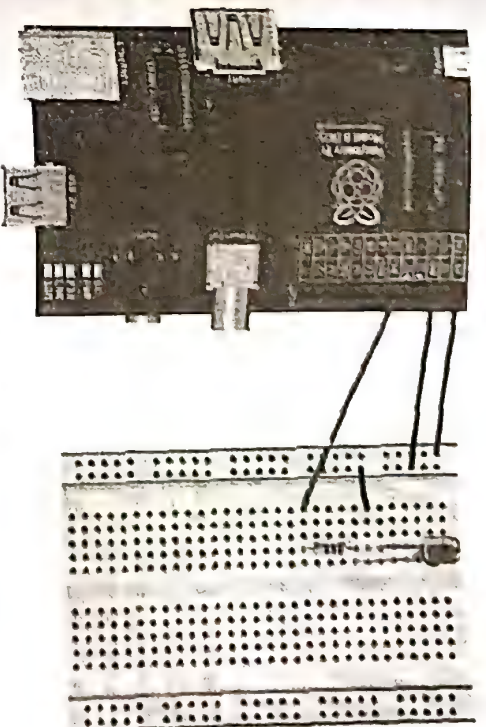
Fig. 5.11 Controlling LED with Raspberry Pi

Q.20. Discuss about interfacing a light sensor with Raspberry Pi.

Ans. Code in Python Language for Switching LED/Light Based on Reading LDR Reading

```
import RPi. GPIO as GPIO
import time
GPIO.setmode (GPIO.BCM)
/dr_threshold = 1000
LDR_PIN = 28
LIGHT_PIN = 26
def readLDR (PIN):
reading = 0
GPIO.setup (LIGHT_PIN, GPIO.OUT)
GPIO.output (PIN, False)
time.sleep (0.1)
GPIO.setup (PIN, GPIO.IN)
while (GPIO.input (PIN) = = False) :
reading = reading + 1
return reading
def switchOnLight (PIN) :
GPIO.setup (PIN, GPIO.OUT)
GPIO.output (PIN, True)
def switchOffLight (PIN) :
GPIO.setup (PIN, GPIO.OUT)
GPIO.output (PIN, False)
```

```
while True :
    ldr_reading = readLDR (LDR_PIN)
    if ldr_reading < ldr_threshold :
        switchOnLight (LIGHT_PIN)
    else :
        switchOffLight (LIGHT_PIN)
    time.sleep (2)
```

Connection of LDR to Raspberry Pi is shown in fig. 5.13. One side of LDR is connected with a to 3.3V and other side to a 1 μF capacitor and also connected with a GPIO pin. An LED is connected to pin 28 which is controlled depended on the light-level sensed. The code for LDR is done by Python code, in which readLDR ( ) function returns a count which is proportional to the light level. The LDR pin is set to output and low and then to input in this function. The capacitor at this point starts charging through the resistor till the input pin reads high. When the input reads high the counter is stopped. The light level depends upon the final count. When the amount of light is greater, lower is the LDR resistance and higher is the time taken by the capacitor for charging at this point starts.

Fig. 5.13 Interfacing LDR with Raspberry Pi

Q.21. Explain in brief about Raspberry Pi 3. Give its applications.

Ans. In February 2016, the single board SoC dependent computing and communication board is Raspberry Pi 3 model B. On the board, an RPi oper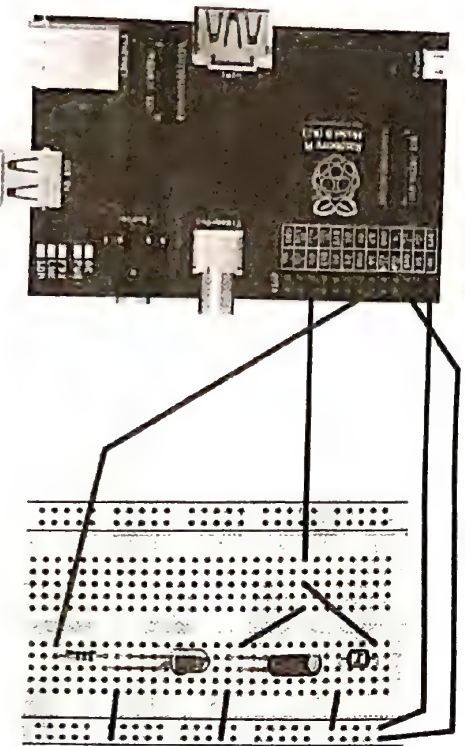ates on the OSes like RISC OS, Windows 10 IoT core, NetBSD, FreeBSD, Plan 9, AROS, Inferno and additional distributions of Linux like Raspbian Ubuntu. The RPi has hardware and software which are used for high performance computing and graphics. Raspberry Pi (RPi) is for devices which require an

operating system like Ubuntu core for home automation and drones. The stripped down version of Ubuntu is the core. Core is developed to operate securely on autonomous machines, M2M and IoT devices. The ARM cortex quad core processor and graphic processor are used in SoC at RPi for graphics and video. It is required input power of 4W, memory on-board of 1 GB, SD card (model B) or micro SDHC card (model B+) slot in case of external SD and micro SD cards. RPi cannot be used for no real-time clock (RTC). Hence an external chip can be used for comprising the RTC.

Applications – RPi board can be used as a personal computer (PC). RPi is used in media server IoT devices. It is also employed in networked security camera systems in home automation on ATMs application and services.

In case of industrial IoT applications, RPi is used Echelon. For the core requirements, the Echelon IIoT platform comprises software in case of IoT which comprise the REST APIs for load control devices, security, wired or wireless devices, web connectivity, autonomous control, physical sensors data collection and transmit to the other devices.

Q.22. Draw the architecture of Raspberry Pi. Write the IoT application for Arduino R-3, Raspberry Pi.

(R.G.P.V., May 2018)

Ans. Refer Q.14, Q.7 and Q.21.

Q.23. Explain Raspberry Pi model B. What are the things need to be considered for developing on the Raspberry Pi ?   (R.G.P.V., Nov. 2019)

Ans. Refer to Q.21 and Q.15.

Q.24. What are the features which makes RPi boards widely used ?

Ans. There are following feature which makes RPi boards widely used as –

(i)   Computer as prototyping which makes them ideal, for developing media server and home or ATM surveillance system IoT applications and services.

(ii)   Supports to coding in Python, C++ and the libraries.

(iii)   With the help of external keyboard and display monitor, software rms on multiple environments and various OSes.

(iv)   Flexible and simple to connect the hardware to external systems.

(v)   A micro-SD slot is used to connect for extended memory.

(vi)   SPI, UART, 12C, 40 GPIO pins are used for extended interfacing capabilities. WiFi module, stereo audio, video with Pi camera module, and stream of high definition HDMI output are also supported by RPi boards.

Q.25. Give the comparison of different IoT devices.

Ans. Table 5.4 shows the comparison of different IoT devices.

Table 5.4

	Raspberry Pi	pcDuino	BeagleBone Black	Cubieboard
CPU	700 MHz ARM1176JZ-F Processor	1 GHz ARM Cortex-A8	AM 335x 1 GHz ARM Cortex-A8	Dual core 1 GHz ARM Cortex-A7
Audio	3.5 mm jack, HDMI	HDMI	HDMI	HDMI
Video	Composite RCA HDMI	HDMI	HDMI	HDMI
Power	5VDC/700 mA	5V/2A	5VDC/460 mA	5VDC/2A
GPU	Dual core video core IV multimedia co-processor	Mali 400	Power VR SGX530	Dual core ARM Mali 400 MP2
Memory	512 MB	1 GB	512MB	1 GB
OS	Rasbian, Pidora, RISC OS, Arch Linux	Ubuntu, Android	Angstron Linux,Android, Ubuntu	Android, Official Linux distribution
Interfaces	GPIO, SPI, I2C, serial	Serial,ADC, PWM, GPIO, serial, CAN, I2C, SPI	69 pin GPIO, SPI, I2C, 4 serial, CAN, GPMC, AIN MMC, XDMA	96 extend pin interface, including I2C,SPI,RGB/ LVDS,CSI/ TS, FM-IN, ADC, CVBS, VGA, SPDIF-OUT, R-TP
Storage	—	2GB flash	2GB on-board flash storage	4 GB NAND flash
Input/Output	2USB, SD, MMC, SDIO card slot	—	4 + 1 USB, MicroSD slot	2 USB, Micro-SD slot,SATA, IR sensor
Network-ing	10/100 M Ethernet	10/100 M Ethernet	10/100 M Ethernet	10/100 M Ethernet

Q.26. Give the differences between Arduino Due and Raspberry Pi model B.

(R.G.P.V., Nov. 2019)

Ans. Table 5.5 shows the differences between Arduino Due and Raspberry Pi model B.

Table 5.5

S.No.	Parameters	Arduino Due	Raspberry Pi B+
(i)	Processor	ATMega328P	Quad-core ARM Cortex A53 Broadcom VideoCore IV with 400 MHz
(ii)	GPU	—	Broadcom VideoCore IV
(iii)	Operating voltage	5 V	5 V
(iv)	Clock speed	16 MHz	1.2 GHz
(v)	System memory	2 kB	1 GB
(vi)	Flash memory	32 kB	—
(vii)	EEPROM	1 kB	—
(viii)	Communication supported	IEEE 802.11 b/g/n IEEE 802.15.4 433RF BLE 4.0 via shield	IEEE 802.11 b/g/n IEEE 802.15.4 433RF BLE 4.0 Ethernet serial
(ix)	Development environments	Arduino IDE	Any linux compatible IDE
(x)	Programming language	Wiring	Python, C, C++, Java Scratch Ruby
(xi)	I/O connectivity	SPI I2C UART GPIO	SPI DSI UART SDIOCSI GPIO

Q.27. Explain in detail about digital service cloud IoT platform.

Ans. Digital service cloud (DSC) is an open IoT platform. It allows IoT innovators to own their customers the way customers own their products. It supports product start-ups, global technological brands and product innovators. It is a platform that accelerates time to market process for a new innovation. By using the readymade infrastructure provided by digital service cloud, one can build needed customised IoT solutions by connecting devices and using plug and play dashboard. It also allows to monitor and manage the product over its lifetime by connecting it with a network of millions of devices. It runs a UI-driven rules engine that requires no coding. It monitors and streams a petabyte of real time data. It operates on a secure tenant based system and provides quick launch of our application with a wizard based app builders.

DSC IoT foundation is shown in fig. 5.14. Here consumer devices equipped with variety of sensors can connect to IoT application available on cloud and can communicate with application. Various networking and connectivity techniques are available that can connect the devices over cloud. Cloud provides device management and app development. Data analytics and visualisation through powerful dashboard are supported to end users.
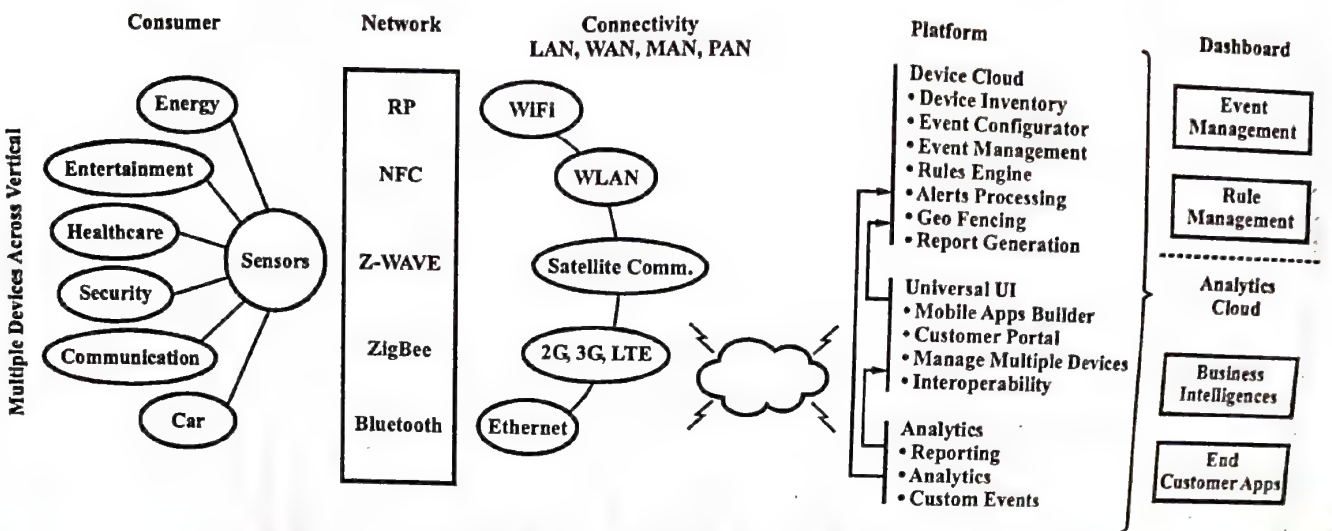
Fig. 5.14 Digital Service Cloud IoT Foundation

Q.28. What are the characteristics of DSC ?

Ans. The characteristics of DSC are as follows —

(i) Scalability

(ii) Security

(iii) Plug and play

(iv) Unified customer service

(v) Users across diverse verticals

(vi) Monitoring, maintenance and management of devices

(vii) Preventive support through support cloud

(viii) Big data analytics

(ix) Multi-channel support delivery.

Q.29. Discuss about the Zetta IoT platform and its features.

Ans. Zetta is an open source platform. It is developed in Node.js for creating IoT servers that run across geo-distributed servers and cloud. Zetta combines REST API, reactive programming and web sockets. This combination is suitable for assembling many devices into data intensive and real time applications. Zetta runs everywhere – in the cloud, on PCs and on single board computers. Raspberry Pis, BwangleBones and PCs together can be linked with cloud platform like Heroku with the help of Zetta so that it can help to create geo-distributed networks. Zetta has the ability to turn any device into an API. By communicating with microcontrollers like Arduino and Spark Core, Zetta can provide every device a REST API as locally and also in the cloud. It provides support for assembling distributed system of devices that communicate and react via APIs. Zetta is a very developer friendly. It gives direct access of underlying protocols and conventions to developers so that they can easily and efficiently transform sensors, actuators and controllers into innovative IoT applications and systems. Architecture of Zetta is optimized for data intensive and real-time applications. Zetta allows monitoring of device and system behaviour in code and using visualization tools to get actionable insights. It also provides streaming of data into machine analytics platforms like Splunk IoT projects consist of multiple devices across multiple locations running multiple applications developed by various companies. Thus, Zetta allows to assemble smartphone apps, device apps and cloud apps together into large and complex systems. Example of such complex and large scale systems include home automation, smart transportation and wearable computing.
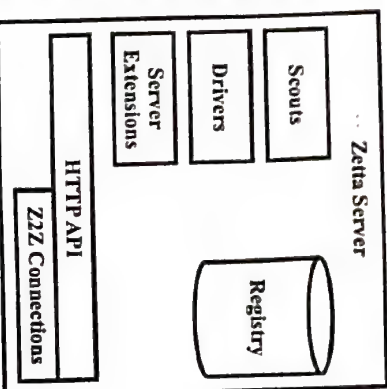

Fig. 5.15 Zetta Server Architecture

Zetta server architecture is shown in fig. 5.15 Zetta server which is at highest level of abstraction contains elements like drivers, scouts, server extension, apps etc. It runs on a hardware hub like Raspberry Pi and allows components to communicate with devices. It then produces API to the consumer. Here Scout is a device discovery technique over network. One Zetta server will reside on hardware hub and another on cloud. Server on hardware will contact to that in the cloud. So Zetta will provide API to consumers.

Features of Zetta are as follows –

(i) Provides API to everything

(ii) Runs everywhere

(iii) Development support of large applications for efficient development

(iv) Provides developer friendly environment

(v) Streaming of big data along with and data insights.

Q.30. Explain the following platforms –

(i) Yaler IoT platform

(ii) Nimbits IoT platform

(iii) Axeda IoT platform.

Ans. (i) Yaler IoT Platform – Yaler is a pay-as-you use platform which provides a relay infrastructure which offers secure access to embedded systems and it works with any device with a TCP socket. Yaler is a cost-effective solution which is suitable for enterprise applications. It also offers a hosted service on dedicated relay instances. A fixed yearly fee per relay instance helps to connect maximum number of devices to a dedicated instance. The Yaler relay service supports SSL/TLS encryption. The integration of Yaler.net, a cloud-based connectivity service, enabled a secure ad-hoc connection for data streaming. It is mainly designed to provide a stable, secure, and high performance execution environment for applications running on Amazon EC2.

(ii) Nimbits IoT Platform – Nimbits is a PaaS that can be downloaded on any device from Raspberry Pi, Web server, Amazon EC2, or Google App Engine. The platform is used for developing hardware and software solutions that can connect to the cloud or to each other and enables logging and retrieving large amounts of data from physical devices, triggering events or alerts, or performance of complex analysis. Nimbits is a platform for connecting people, sensors and software to the cloud and with one another. It is based upon data logging and rule based technology. It resolves the complexity associated with Edge Computing in IoT by facilitating a platform which is built upon embedded system locally and then filtering noise, running rule engine and then pushing data that are very important on the cloud. It first records and then processes geo informatics and time stamped data and then produces rule form that information. Here rules can be email alerts, push notification, statistics or any calculation. Nimbits Public Cloud is an instance of Nimbits server.

(iii) Axeda IoT Platform – The Axeda Machine Cloud service provides M2M services, IoT connectivity services, software agents and toolkits that allow us to select communication method and hardware which is appropriate for our IoT solution. The Axeda IoT platform provides end-to-end enterprise capabilities using Java EE technology and the Oracle database. It gives facilities like message queuing for reliable end-to-end data transfer, security based on SSL communication with authentication and authorization, device and asset management, tracking and monitoring, complex event processing, location services, and more. Oracle's Java Embedded solutions aim to support massive amounts of data required for and created as a result of the Internet of Things by facilitating seamless communications between all elements of the IoT architecture. It delivers an integrated, secure and comprehensive platform for the entire IoT architecture across all vertical markets. Oracle enables real-time response and data capture from millions of device endpoints. Oracle offers several solutions, including Oracle Java SE Embedded, Oracle Java ME Embedded, Oracle Java Embedded Suite and Oracle Event Processing for Oracle Java Embedded to meet any specific technology requirements.

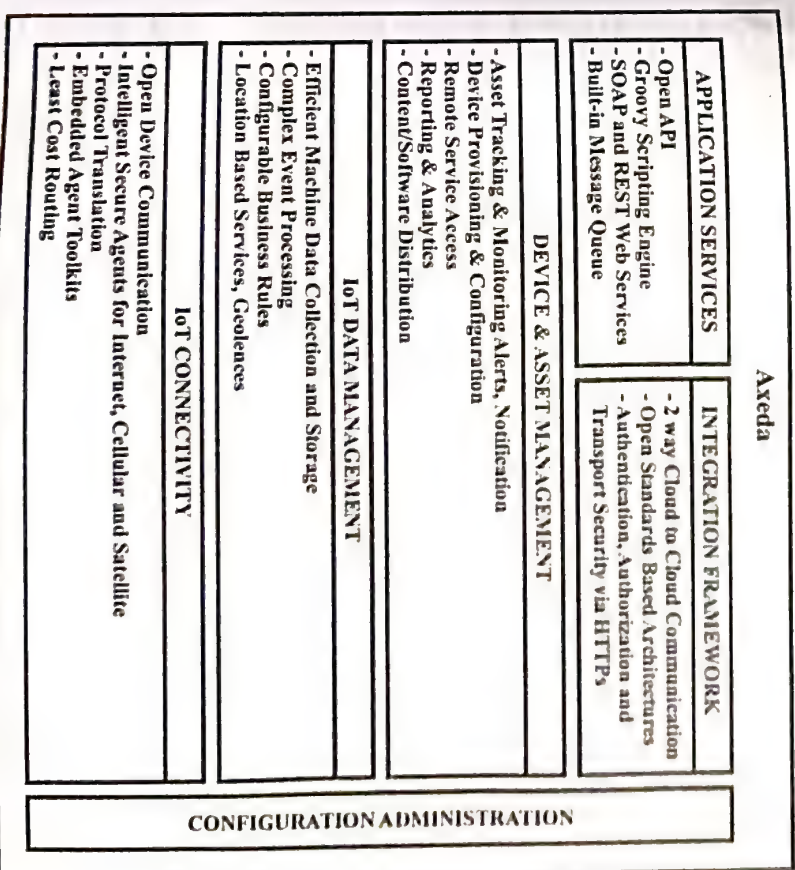| Axeda	
APPLICATION SERVICES	**INTEGRATION FRAMEWORK**
- Open API - Groovy Scripting Engine - SOAP and REST Web Services - Built-in Message Queue	- 2 way Cloud to Cloud Communication - Open Standards Based Architectures - Authentication, Authorization and Transport Security via HTTPs
DEVICE & ASSET MANAGEMENT	
- Asset Tracking & Monitoring Alerts, Notification - Device Provisioning & Configuration - Remote Service Access - Reporting & Analytics - Content/Software Distribution	
IoT DATA MANAGEMENT	
- Efficient Machine Data Collection and Storage - Complex Event Processing - Configurable Business Rules - Location Based Services, Geolences	
IoT CONNECTIVITY	
- Open Device Communication - Intelligent Secure Agents for Internet, Cellular and Satellite - Protocol Translation - Embedded Agent Toolkits - Least Cost Routing	
CONFIGURATION ADMINISTRATION	

Fig. 5.16 Axeda IoT Platform

Axeda IoT platform stack is shown in fig. 5.16 IoT connectivity is at bottom level. Above it IoT data management layer is there to manage data coming from devices. Devices and assets are managed by using cloud platform. At the top level application service and integration framework support application development and deployment. Device and asset configuration management is also supported by cloud.

Q.31. Write down the features of Yaler, Nimbits and Axeda platforms.

Ans. Features of Yaler IoT platform are –

(i) Access via any browser or even phone

(ii) It provides plug-and-play functionality for end users

(iii) It provides access to your device from a client such as a browser, firewalls, NAT or mobile routers.

(iv) Enables addressability and accessibility for devices blocked by android, etc.

Features of Nimbits IoT platform are –

(i) Compatible with most J2EE servers

(ii) Build provided for Google App Engine and Linux Systems for development

(iii) Process geo and time-stamped data

(iv) Download Nimbits servers on chips, servers or on the cloud

(v) It is an Open-source platform

(vi) Facilitates event triggers and alerts.

Features of Axeda platform are –

(i) Facilitates communications among devices

(ii) It is an M2M platform with proven middleware capabilities

(iii) Real time data capture

(iv) Millions of device.

DATA ANALYTICS FOR IoT, CLOUD FOR IoT, CLOUD STORAGE MODELS & COMMUNICATION APIs

---

Q.32. What is big data ? Explain.

Ans. "Data of a very large size and typically to the extent that its manipulation and management present significant logistical challenges" is known as big data.

The technologies and initiatives that involve data that is too diverse, fast-changing or massive for conventional technologies, skills and infrastructure to address efficiently is referred to as big data. The data sets that are so large, complex, and impractical to manage with traditional software tools are described by big data.

But now the information from big data can be analyzed by using new technologies e.g., user web clicks can be tracked by retailers to identify behavioural trends that improve campaigns, pricing and stockage.

Major web companies such as Google, Amazon, and Facebook pioneered businesses built on monetizing massive data volumes over the last decade. The new paradigms not only for extracting value from data but also for managing data and compute resources from data center design, to hardware, to software, to application provisioning were invented by them.

Another definition of big data is as follows –

"The collection, processing, discovery, analysis and storage of large volumes and disparate types of data is enabled by the emerging technologies and practices, very quickly and cost effectively".

Q.33. Explain big data types with examples.

Ans. Big data encompasses everything, from dollar transactions to tweets to images to audio. Therefore, taking advantage of big data requires that all this information to be integrated for analysis and data management. This is more difficult than it appears. Big data includes huge volume, high velocity, and extensible variety of data. There are three types of data concerned here –

(i) Structured Data – This is the data stored in relational databases table in the format of row and column. They have fixed structures and these structures are defined by organizations by creating a model. The model allows to store, process as well as gives permission to operate the data. The model defines the characteristics of data including data type and some restrictions on the data. Analysis and storing of structured data is very easy. Because of high cost, limited storage space and techniques used for processing, causes RDBMS the only path to store and process the data effectively. Programming language called structured query language (SQL) is used for managing this type of data.

(ii) Semi-structured Data – Data which is in the form of structured data but does not fit the data model is semi-structured data. It cannot be stored in the form of data table, but it can be stored in some particular types of files which hold some specific markers or tags. These markers are distinguished by some specific rule and the data is enforced to be stored with a ranking. This form of data increased rapidly after the introduction of the World Wide Web where various form of data need medium for interchanging the information like XML and JSON.

Example – CSV, XML, XML and JSON documents are semi-structured documents, NoSQL databases are considered as semi-structured.

(iii) Unstructured Data – Data without any specific structure and due to this could not be stored in a row and column format is unstructured data. This data is contradictory to that of structured data. It cannot be stored in a databank. Volume of this data is growing extremely fast which is very tough to manage and analyze it completely. To analyze the unstructured data advanced technology knowledge is needed.

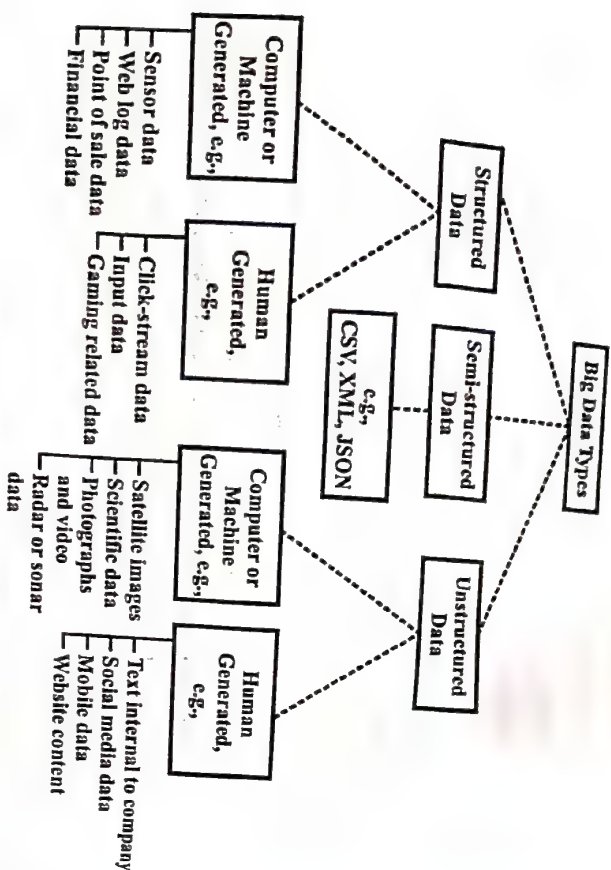Fig. 5.17, depict three types of big data along with examples.

Fig. 5.17 Big Data Types

Big Data Types
- Structured Data
  - Computer or Machine Generated, e.g.,
    - Sensor data
    - Web log data
    - Point of sale data
    - Financial data
  - Human Generated, e.g.,
    - Click-stream data
    - Input data
    - Gaming related data
- Semi-structured Data
  - e.g., CSV, XML, JSON
- Unstructured Data
  - Computer or Machine Generated, e.g.,
    - Satellite images
    - Scientific data
    - Photographs and video
    - Radar or sonar data
  - Human Generated, e.g.,
    - Text internal to company
    - Social media data
    - Mobile data
    - Website content

Q.34. Discuss 5V characteristics of big data.

Ans. Big data is important because it enables organizations to gather, store, manage, and manipulate vast amounts of data at the right speed, at the right time, to gain the right insights. In addition, Big data generators must create scalable data (volume) of different types (variety) under controllable generation rates (velocity), while maintaining important characteristics of the raw data (veracity), the collected data can bring to the intended process, activity or predictive analysis/hypothesis. Indeed, there is no clear definition for 'Big Data'. It has been defined based on some of its characteristics. Therefore, these five characteristics have been used to define Big Data, earlier known as 4V's (volume, variety, velocity and veracity), as illustrated in fig. 5.18.
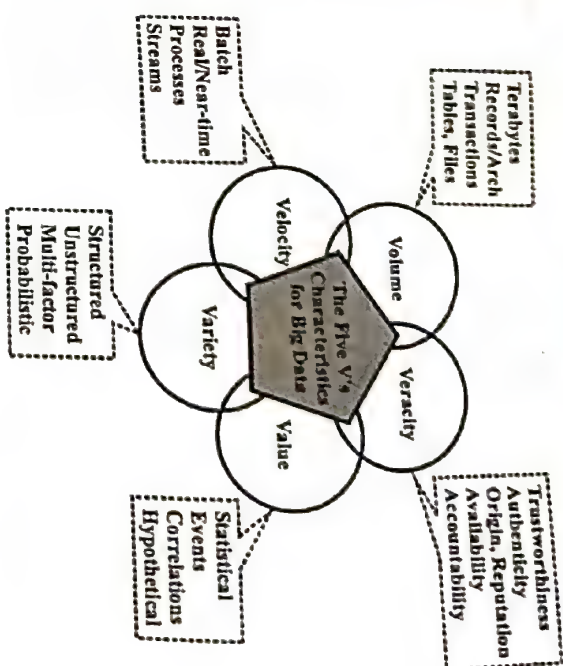
- Volume: Terabytes, Records/Arch, Transactions, Tables, Files
- Velocity: Batch, Real/Near-time, Processes, Streams
- Variety: Structured, Unstructured, Multi-factor, Probabilistic
- Veracity: Trustworthiness, Authenticity, Origin, Reputation, Availability, Accountability
- Value: Statistical, Events, Correlations, Hypothetical

Fig. 5.18 Five V's Characteristics of Big Data

(i) Volume – It refers to the quantity of data gathered by a company. This data must be used further to gain important knowledge. Enterprises are awash with ever-growing data of all types, easily amassing terabytes even petabytes of information (e.g., turning 12 terabytes of tweets per day into improved product sentiment analysis; or converting 350 billion annual meter readings to better predict power consumption).

Moreover, Demchenko, Grosso, de Laat and Membrey stated that volume is the most important and distinctive feature of Big Data, imposing specific requirements to all traditional technologies and tools currently used.

(ii) Velocity – It refers to the time in which Big Data can be processed. Some activities are very important and need immediate responses, which is why fast processing maximizes efficiency. For time-sensitive processes like fraud detection, Big data flows must be analyzed and used as they stream into the organizations in order to maximize the value of the information (e.g., scrutinize 5 million trade events created each day to identify potential fraud; analyze 500 million daily call detail records in real-time to predict customer churn faster).

(iii) Variety – It refers to the type of data that big data can comprise. This data may be structured or unstructured. Big data consists different types of data, including structured and unstructured data such as text, sensor data, audio, video, click streams, log files and so on. The analysis of combined data

types brings new problems, situations, and so on, such as monitoring hundreds of live video feeds from surveillance cameras to target points of interest, exploiting the 80% data growth in images, video and documents to improve customer satisfaction.

(iv) Value – It refers to the important feature of the data which is defined by the added-value that the collected data can bring to the intended process, activity or predictive analysis/hypothesis. Data value will depend on the events or processes they represent such as stochastic, probabilistic, regular or random. Depending on this the requirements may be imposed to collect all data, store for longer period (for some possible event of interest), etc. In this respect data value is closely related to the data volume and variety.

(v) Veracity – It refers to the degree in which a leader trusts information in order to make a decision. Therefore, finding the right correlations in Big Data is very important for the business future. However, as one in three business leaders do not trust the information used to reach decisions, generating trust in big data presents a huge challenge as the number and type of sources grows.

Q.35. What do you mean by big data analytics ? Explain various types of analytics.

Ans. Big data analytics, is the process of examining large data sets that containing a variety of data types i.e., big data to uncover all hidden patterns, unknown correlations, market trends, customer preferences and other useful business information. Then analytical findings can lead to more effective marketing, new revenue opportunities, better customer service, improved operational efficiency, competitive advantages over rival organizations and other business benefits.

The primary goal of big data analytics is to help companies make more informative business decisions by enabling data scientists, predictive modellers and other analytics professionals to analyse large volumes of transactional data, as well as other forms of data that may be untapped by more conventional business intelligence (BI) programs. That could include web server logs and Internet click stream data, social media content and social network activity reports, text from customer e-mails and survey responses, mobile phone call detail records and machine data captured by sensors and connected to the Internet of Things.

Big data burst upon the scene in the first decade of the 21st century, and the first organizations to embrace it were online and start-up firms. Arguably, firms like Google, LinkedIn, eBay and Facebook were built around big data from the beginning. They did not have to reconcile or integrate big data with more traditional sources of data and the analytics performed upon them, because

they did not have that much of traditional forms. They did not have to merge big data technologies with their traditional IT infrastructures because these infrastructures did not exist. Big data could stand alone, big data analytics could be the only focus of analytics, and big data technology architectures could be the only architecture. So big data using Hadoop and No SQL free software's.

Analytics can be classified into following three types –

  (i) Predictive analytics

  (ii) Descriptive analytics

  (iii) Prescriptive analytics.

(i) Predictive Analytics – Predictive analysis establish previous data patterns and gives list of solutions which may come for given situation. Predictive analysis study the present as well as past data and predict what may happen in future, give probabilities of what would happen. It is used to your big data to forecast other data which we do not have. This analytical method is one of the most commonly used methods used for sales lead scoring, social media and consumer relationship management data.

Three basic elements of predictive analytics are as follows –

    (a) Predictive modelling

    (b) Decision analysis and optimization

    (c) Transaction profiling.

For example, predictive analytics is used for optimizing customer relationship management systems. They can help enable an organization to analyze all customer data therefore exposing patterns that predict customer behaviour.

(ii) Descriptive Analytics – Descriptive analytics also known as data mining, operates what is happening in real-time. It is one of the simplest types of analytics as it converts big data into small bytes. The result is monitored through e-mails or dashboard. It is used by majority of organizations.

For example, descriptive analytics examines historical electricity usage data to help power needs and allow electric companies to set optimal prices.

(iii) Prescriptive Analytics – Prescriptive analytics reveals actions and recommend of what step should be taken. It gives answer to the situation in a focused way. Prescriptive data analytics goes one step forward of predictive as it provides multiple actions with likely outcomes for each decision. This method of analytics is not preferred much by organizations, but its data can show impressive result if used correctly.

For example, prescriptive analytics can benefit healthcare strategic planning by using analytics to leverage operational and usage data combined with data

of external factor such as economic data, population demographic trends and population health trends, to more accurately plan for future capital investments such as new facilities and equipment utilization as well as understand the trade-offs between adding additional beds and expanding an existing facility versus building a new one.

Q.36. Explain core components of analytical data architecture.

Ans. The big data storage and analytics platform provides resources and functionalities for storage as well as for batch and real-time processing of the big data. It provides main integration interfaces between the site operational platform and the cloud data lab platform and the programming interfaces for the implementation of the data mining processes. The internal structure of the big data storage and analytics platform is given in fig. 5.19.
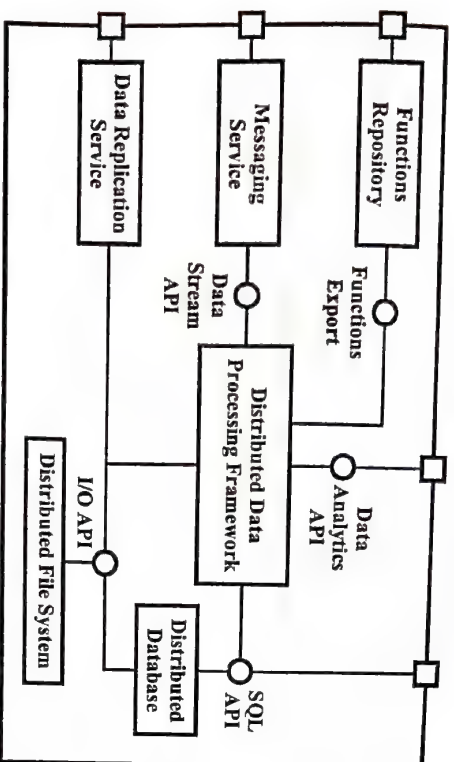

Fig. 5.19 The Internal Architecture of the Big Data Storage and Analytics Platform

Data are primarily stored in the distributed file system, which is responsible for the distribution and replication of large datasets across the multiple servers (data nodes). A unified access to the structured data is provided by the distributed database using the standard SQL interface. The main component responsible for data processing is the distributed data processing framework, which provides high-level API for the implementation of the data pre-processing tasks and for the building and validation of the predictive functions. Predictive functions are stored in the functions repository, where they are available for production deployment or for the simulations and overall optimization of the production processes. The rest of the components (messaging service and data replication service) provide data communication interfaces, and connect the operational platform to the data lab platform.

The big data storage and analytics platform consist of the following sub-components –

(i) *Distributed File System* – Provides a reliable, scalable file system with similar interfaces and semantics to access data as local file systems.

(ii) *Distributed Database* – Provides a structured view of the data stored in the data lab platform using the standard SQL language, and supports standard RDBMS programming interfaces such as JDBC for Java or ODBC for .Net platforms.

(iii) *Distributed Data Processing Framework* – Allows the execution of applications in multiple nodes in order to retrieve, classify or transform the arriving data. The framework provides data analytics APIs for two main paradigms for processing large datasets – API for parallel computation and API for distributed computation.

(iv) *Functions Repository* – Provides storage for predictive functions together with all settings required for the deployment of functions.

(v) *Messaging Service* – Implements an interface for real-time communication between the data lab and operation platforms. It provides a publish-subscribe messaging system for asynchronous real-time two-way communication, which allows to the decoupling of data providers and consumers.

(vi) *Data Replication Service* – Provides an interface for uploading of the historical batch data between the data lab and operation platform.

Q.37. Explain the application of big data analytics in various fields.

Ans. Big data analytics applications (BDA Apps) are a new category of software applications that leverage largescale data, which is typically too large to fit in memory or even on one hard drive, to uncover actionable knowledge using large scale parallel-processing infrastructures. The big data can come from sources such as runtime information about traffic, tweets during the Olympic Games, stock market updates, usage information of an online game, or the data from any other rapidly growing data intensive software system.

(i) *In Clustering* – Using clustering (K-means algorithm) through a simple point and click dialog, users can automatically find groups within data based on specific data dimensions. With clustering, it is then simple to identify and address groups by customer type, text documents, products, patient records, click path, behaviour, purchasing patterns, etc.

(ii) *In Data Mining* – Datameer's decision trees automatically help users understand what combination of data attributes result in a desired outcome. Decision trees illustrate the strengths of relationships and dependencies

within data and and are often used to determine what common attributes influence outcomes such as disease risk, fraud risk, purchases and online signups. The structure of the decision tree ref lects the structure that is possibly hidden in big data.

(iii) In Banking – The use of customer data invariably raises privacy issues. By uncovering hidden connections between seemingly unrelated pieces of data, big data analytics could potentially reveal sensitive personal information. Research indicates that 62% of bankers are cautious in their use of big data due to privacy issues. Further, outsourcing of data analysis activities or distribution of customer data across departments for the generation of richer insights also amplifies security risks. For instance, a recent security breach at a leading UK-based bank exposed databases of thousands of customer files. Although this bank launched an urgent investigation, files containing highly sensitive information such as customers' earnings, savings, mortgages, and insurance policies ended up in the wrong hands. Such incidents reinforce concerns about data privacy and discourage customers from sharing personal information in exchange for customized offers.

(iv) In Marketing – Marketers have begun to use facial recognition software to learn how well their advertising succeeds or fails at stimulating interest in their products. A recent study published in the Harvard Business Review looked at what kinds of advertisements compelled viewers to continue watching and what turned viewers off. Among their tools was "a system that analyses facial expressions to reveal what viewers are feeling." The research was designed to discover what kinds of promotions induced watchers to share the ads with their social network, helping marketers create ads most likely to "go viral" and improve sales.

(v) In Smart Phones – Perhaps more impressive, people now carry facial recognition technology in their pockets. Users of iPhone and Android smart phones have applications at their fingertips that use facial recognition technology for various tasks. For example, Android users with the remember app, can snap a photo of someone, then bring up stored information about that person based on their image when their own memory lets them down a potential boon for salespeople, iPhone users can unlock their device with recognize me, an app that uses facial recognition in lieu of a password. If deployed across a large enterprise, this app could save an average of $2.5 million a year in help-desk costs for handling forgotten passwords.

(vi) In Telecom – Now-a-days big data is used in different fields. In telecom also it plays a very good role. Service providers are trying to compete in the cut-throat world of telecom services. Where more and more subscribers rely on over-the-top (OTT) players as providers of value-added services are

focused on increasing revenue, reducing opex, churn and enhancing the customer experience as key business objectives.

Operators believe that big data and advanced analytics will play a critical role in helping them meet their business objectives. In the same survey, respondents indicate critical use case scenarios in the context of big data and advanced analytics where they are investing now and where they plan to invest in the next three years.

Operators face an uphill challenge when they need to deliver new, compelling, revenue generating services without overloading their networks.

(vii) In Agriculture – A biotechnology firm uses sensor data to optimize crop efficiency. It plants test crops and runs simulations to measure how plants react to various changes in condition. Its data environment constantly adjusts to changes in the attributes of various data it collects, including temperature, water levels, soil composition, growth, output, and gene sequencing of each plant in the test bed. These simulations allow it to discover the optimal environmental conditions for specific gene types.

Q.38. What types of analytics can be performed in IoT ?

Ans. Following types of analytics can be performed in IoT –

(i) Streaming Analytics – This form of data analytics is also referred as event stream processing and it analyzes huge in-motion data sets. Real-time data streams are analyzed in this process to detect urgent situations and immediate actions. IoT applications based on financial transactions, air fleet tracking, traffic analysis etc. can benefit from this method.

(ii) Spatial Analytics – This is the data analytics method which is used to analyze geographic patterns to determine the spatial relationship between the physical objects. Location-based IoT applications, such as smart parking applications can benefit from this form of data analytics.

(iii) Time Series Analytics – As the name suggests, this form of data analytics is based upon the time-based data which is analyzed to reveal associated trends and patterns. IoT applications, such as weather forecasting applications and health monitoring systems can benefit from this form of data analytics method.

(iv) Prescriptive Analysis – This form of data analytics is the combination of descriptive and predictive analysis. It is applied to understand the best steps of action that can be taken in a particular situation. Commercial IoT applications can make use of this form of data analytics to gain better conclusions.

Q.39. Discuss about the impacts of IoT on big data.

Ans. IoT is a network consisting of physical devices, which are also implanted with sensors, electronics, and software, thereby allowing these devices to exchange data. This ultimately allows better incorporation between real world physical entities and computer-operated systems. IoT is the next big thing impacting our lives in major ways and number of factors. Technologies like column-oriented databases, SQL in Hadoop, Hive, Wibidata, PLATFORA, SkyTree, Storage technologies, Schema-less databases, Big Data Lambda Architecture, Map-reduce, Streaming Big Data analytics, Big Data analytics, or NoSQL databases, PIG, etc., helps in dealing with the enormous amount of data generated by IoT and other sources.

The main factors that big data is impacted by IoT are –

(i) Big Data Storage – At basis, the key necessities of big data storage are that it can handle very huge amounts of data and continuous balancing to keep up with expansion and that it can provide the input/output operations per second (IOPS) necessary to deliver data to analytics tools. The data is of different form and format and thus, a datacenter for storing such data must be able to handle the load in changeable forms. IoT has a direct impact on the storage infrastructure of big data. Collection of IoT big data is a challenging task because filtering redundant data is mandatorily required. After collection, the data has to transfer over a network to a data center and maintained. Many companies started to use Platform as a Service (PaaS) to handle their infrastructure based on IT. It helps in developing and running web applications. By this way, big data can be managed efficiently without the need of expanding their infrastructural facilities to some extent. IoT big data storage is certainly a challenging task as the data grows in a faster rate than expected.

(ii) Data Security Issues – The IoT has given new security challenges that cannot be controlled by traditional security methods. Facing IoT security issues require a shift. For instance, how do you deal with a situation when the television and security camera at your home are fitted with unknown Wi-Fi access.

Few security problems are –

(a) Secure computations in distributed environment

(b) Secure data centers

(c) Secure transactions

(d) Secure filtering of redundant data

(e) Scalable and secure data mining and analytics

(f) Access control

(g) Imposing real time security, etc.

A multi-layered security system and proper network system will help avoid attacks and keep them from scattering to other parts of the network. An IoT system should follow rigorous network access control policies and then allowed to connect. Software defined networking (SDN) technologies should be used for point-to-point and point-to-multipoint encryption in combination with network identity and access policies.

(iii) Big Data Analytics – Data analytics is the science of examining raw data with the idea of coming to conclusions about that information. Big data analytics will help you understand the business value it brings and how different industries are applying it to deal with their sole business necessities. According to the Gartner IT dictionary, Big data is variety of information assets, high-volume, and high-velocity and innovative forms of information processing for enhanced approach and decision making.

Here volume refers to the size of data, variety refers to the number of forms of data, velocity refers to the speed of data processing.

Big data analytics is used in many industries to allow them to make better business decisions and in the sciences to verify or disprove existing models or theories. IoT big data analytics is very much needed to end up in a optimized decision.

(iv) Impact on Day to Day Living – IoT big data is slowly redefining our lives. Let us consider a few examples of our lives. At work, the CCTV camera in the canteen estimating the time you spend there. The class room sensors can find out how much time you spend in writing on the board. This can be just to measure the productivity of an employee.

At home, the home theatre playing our favourite movie as soon as we switch on the television, smart devices could save a lot of power and money by automatically switching off electrical devices when we leave home. A smart wrist band tied to the elder people at home intimates the nearby hospital, if they fall sick.

All this and much more is going be real in a very short time because of the rapid development in IoT and Big data technologies.

Q.40. What are the challenges for IoT big data.

Ans. Some of the challenges for IoT big data are as follows –

(i) Huge data volumes

(ii) Difficulty in data collection

(iii) Incompatible standards

(iv) New security threats

(v) No reliability in the data

(vi) Fundamental shifts in business models

(vii) Huge amount of data to analyze

(viii) A rapidly evolving privacy landscape.

Q.41. Explain big data technologies.

Ans. Using modern computing technology, businesses may now manage immense volumes of data previously could dealt with using expensive supercomputers. These are now much cheaper. As a result, new techniques for distributed computing are main stream. Big data became paramount as companies such as Yahoo!, Google, and Facebook came to the realization that they required help in monetizing the massive amounts of data their offerings were creating. Thus, these new companies must search for new technologies to store, access, and analyze huge amounts of data in near real time. Such real-time analysis is required in order to profit from so much data from users. Their resulting solutions have affected the larger data management market. In particular, the innovations MapReduce, Hadoop, and Big Table have proven lead to a new generation of data management. These technologies will allow businesses to address one of the most fundamental problems, namely the compability to process massive amounts of data efficiently, cost-effectively, and quickly.

(i)   MapReduce – MapReduce was designed by Google to efficiently carry out a set of functions against a large amount of data in batch mode. The "map" component distributes the programming problem or task across a large number of systems while managing placement to balance the load and allow recovery from failures. After the distributed computation is complete, another function called "reduce" aggregates all the elements back together to provide a result. An example of MapReduce would be determining the number of pages in a book that are written in each of 50 different languages.

(ii)   Big Table – Big Table was developed by Google as a distributed storage system to manage highly scalable structured data. Data is organized into tables with rows and columns. Unlike typical relational database models, Big Table is a sparse, distributed, persistent multidimensional sorted map. It has been designed to keep large volumes of data across commodity servers.

(iii)   Hadoop – Hadoop is an Apache-managed software framework created using MapReduce and Big Table. Hadoop allows applications based on MapReduce to run on large clusters of commodity hardware. The project has become the basis for the computing architecture underlying Yahoo!'s business. Hadoop is designed to parallelize data processing across computing nodes to speed computations and diminish latency. Two major components of Hadoop exist – a massively scalable distributed file system that can support petabytes of data, and a massively scalable MapReduce engine that computes results in batches.

Q.42. Explain main components of Hadoop.

Ans. Two main components of Hadoop are as follows –

(i)   The Hadoop Distributed File System (HDFS) – HDFS is the storage system for a cluster. When data lands in the cluster, HDFS breaks it into pieces and distribute those pieces among the different servers participating in the cluster. Each server stores just a small fragment of the complete data set and each piece of data is replicated on more than one server.
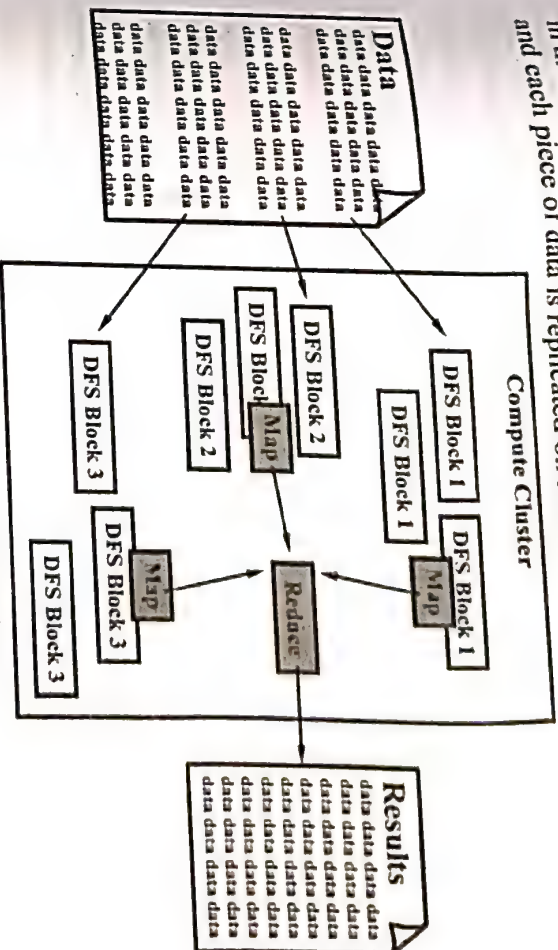

Fig. 5.20 HDFS & MapReduce

(ii)   MapReduce – Because Hadoop stores the entire data set in small pieces across a number of servers, analytical jobs can be distributed in parallel to each of the servers storing part of the data. Each server evaluates the question against its local fragment simultaneously and reports its result back for collation into a comprehensive answer. MapReduce is the agent that distributes the work and collects the results. Both HDFS and MapReduce are designed to continue to work even if there are failures. HDFS continuously monitors the data stored on the cluster. If a server becomes unavailable, a disk drive fails or data is damaged due to hardware or software problems, HDFS automatically restores the data from one of the known good replicas stored elsewhere on the cluster. MapReduce monitors the progress of each of the servers participating in the job, when an analysis job is running. If one of them is slow in returning an answer or fails before completing its work, MapReduce automatically starts another instance of the task on another server that has a copy of the data.

Because of the way that HDFS and MapReduce work, Hadoop provides scalable, reliable and fault-tolerant services for data storage and analysis at very low cost.

Q.43. Explain the ecosystem of Hadoop.

Ans. Hadoop is an open source framework maintained by the Apache Foundation for reliable, scalable and distributed computing. According to the website hadoop.apache.org, the components of Hadoop are defined as projects which function different to each other's. Some of the widely used Hadoop components are as follows –

(i) **Pig** – It is a platform for HDFS. It consists of a compiler for MapReduce programs and a high-level language called Pig Latin. It provides a way to perform data extractions, transformations and loading, and basic analysis without having to write MapReduce programs.

(ii) **Hive** – It is a distributed data warehouse. A data warehouse and SQL-like query language that presents data in the form of tables. Hive programming is similar to database programming. (It was initially developed by Facebook).

(iii) **HBase** – It is a non-relational, distributed database that runs on top of Hadoop. HBase tables can serve as input and output for MapReduce jobs.

(iv) **Zookeeper** – It is an application that coordinates distributed processes.

(v) **Mahout** – Mahout is a data mining software that can be easily scalable. Mahout offers java libraries or scalable machine learning algorithm which can be used for analyzing the data. These machine learning algorithms allow user to perform a task such as classification, clustering, association rule analysis, and predictive analysis.

(vi) **Cassandra** – Hadoop Cassandra provides database that can be easily scalable and highly available without interruption in the job performance.

(vii) **Chukwa** – Chukwa is a data collections system which is mainly used for displaying, monitoring, and analyzing the outcomes of the collected data.

(viii) **Spark** – Spark is a computing system which is used for configuring the Hadoop cluster for fast processing of Hadoop data. Spark does not use MapReduce job of execution engine to run the job. It uses its own distributed runtime to complete the job.

(ix) **Tez** – Tez is a data-flow programming language build in the Hadoop Yarn to execute an arbitrary DAG of tasks to process data for both batch and interactive use-case.

(x) **Avro** – Avro is used for data serialization which provides a container file for storing persistent data. Avro was created by Doug Cutting for making Hadoop to be writable in many programming languages such as C, C++, C#, Java, JavaScript, Python, Ruby.

(xi) **Ambari** – It is a web interface for managing, configuring and testing Hadoop services and components.

(xii) **Flume** – It is a software that collects, aggregates and moves large amounts of streaming data into HDFS.

(xiii) **Sqoop** – It is a connection and transfer mechanism that moves data between Hadoop and relational databases.

(xiv) **Oozie** – It is a Hadoop job scheduler.
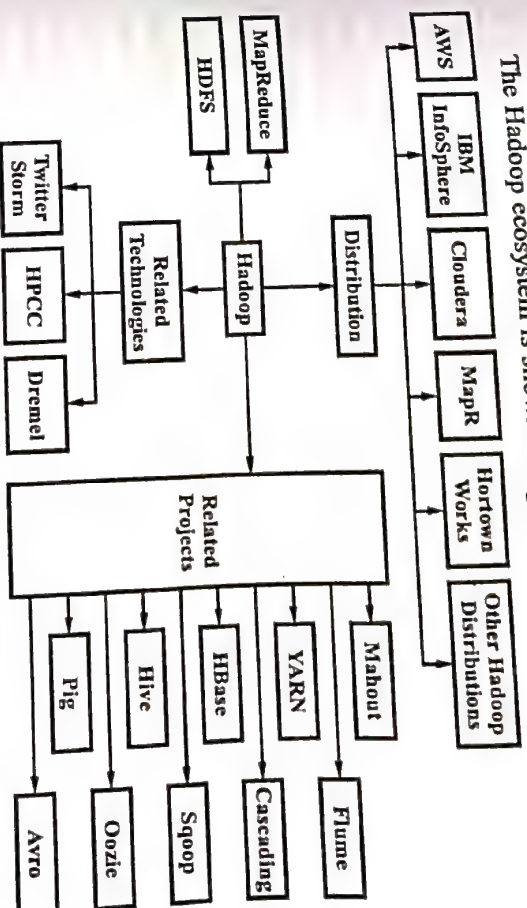
The Hadoop ecosystem is shown in fig. 5.21.

Fig. 5.21 The Hadoop Ecosystem

Q.44. Define cloud computing.

Ans. The term cloud computing refers to the means of providing any and all information technology from computing power to computing infrastructure, applications, business processes and personal collaboration to end users as a service when they require it.

The term cloud in cloud computing means the set of software, hardware, networks, storage, services, and interfaces that combine to provide aspects of computing as a service.

A definition given by American National Institute of Standards and Technology (NIST) is as follows –

Cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (for example, networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction.

A single area of concern in cloud computing is undoubtedly be privacy and security. When your data travels over and rests on systems that are no longer under your control, you have increased risk due to the interception and malfeasance of others. You can't count on a cloud provider maintaining your privacy in the face of government actions.

Q.45. What does Infrastructure-as-a-Service (IaaS) refer to ? Explain.

Ans. Infrastructure/Hardware as a service solutions are the most popular of cloud computing. The available alternatives within the IaaS-offering umbrella are database, Web servers, load balancers and network devices from single servers to entire infrastructures. Hardware virtualization is the main technology used to provide and implement these solutions. In hardware virtualization, one or more virtual machines are suitably configured and interconnected that define the distributed system on top of which applications are installed and deployed. Virtual machines also form the atomic components. These components are deployed and priced depending on the memory, disk storage and number of processors. The advantages of hardware virtualization are sandboxing, hardware tuning, workload partitioning and application isolation. All these advantages of hardware virtualization are provided by IaaS/HaaS solutions. IaaS/HaaS solutions decreases the administration and maintenance cost, and the capital costs from the view point of the customer. It also enables better exploitation of the IT infrastructure and offers a more secure environment for executing third-party applications from the view point of the service provider. Simultaneously, users can take benefit of the full customization provided by virtualization to deploy their infrastructure in the cloud. Virtual machines mostly come with the selected OS installed and the system can be configured with all the needed packages and applications. Apart from the basic virtual machine management capabilities, some other services can be offered. These services are workload management, SLA resource based allocation, ability to integrate third party IaaS solutions, and support for infrastructure design through advanced Web interfaces.

Q.46. Explain Platform-as-a-Service (PaaS) solutions in detail.

Ans. In the cloud, PaaS solutions offer a development and deployment platform for executing applications. PaaS solutions form the middleware on top of which applications are made. The main functionality of the middleware is application management. PaaS implementations automate the process of deploying applications to the infrastructure, provisioning and configuring supporting technologies, configuring applications components, and managing system change on the basis of policies defined by the user. They do not expose any service for managing the underlying infrastructure and offer applications with a runtime environment

of applications and are not related with operating systems; hardware, and other low-level services. According to the commitments done with the users, the core middleware is responsible for managing the resources and scaling applications automatically or on request. The core middleware exposes interfaces that permit programming and deploying applications on the cloud from a user perspective. These can be in the form of programming APIs and libraries or in the form of a Web-based interface. Fig. 5.22 provides an overall view of the PaaS approach.
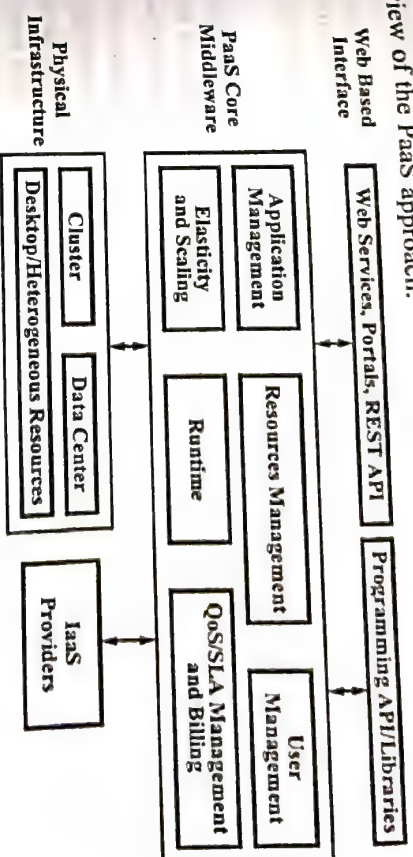
Web Based Interface	Web Services, Portals, REST API	Programming API/Libraries	
PaaS Core Middleware	Application Management	Resources Management	User Management
	Elasticity and Scaling	Runtime	QoS/SLA Management and Billing
Physical Infrastructure	Desktop/Heterogeneous Resources		
	Cluster	Data Center	IaaS Providers

Fig. 5.22 Platform-as-a-Service

The interface exposed to the user is determined by the certain development model decided for applications. Certain implementations offer a fully Web based interface hosted in the cloud offering various services. It is possible to discover integrated developed environments on the basis of 4GL and visual programming concepts, or rapid prototyping environments in which applications are made by assembling mash-ups and user defined components, and successively customized. Other implementations of the PaaS model offer a programming language based approach and offer a complete object model for representing an application. This approach gives more opportunities and flexibility, however, generates longer development cycles. Generally, developers have the full power of programming languages with some limitations to offer better scalability and security. In this situation, the conventional development environments are used to design and develop applications, which are then deployed on the cloud by employing the APIs exposed by the PaaS provider. For better utilizing the services given by the PaaS environment, specific components can be provided together with the development libraries.

Q.47. Describe Software-as-a-Service (SaaS) solution. How it relates to cloud computing ?

Or

What do you understand by SaaS ?

Ans. The concept of SaaS is precedent to cloud computing and started to circulate at the end of 90s. SaaS is a software delivery model which offers access to applications using the Internet. It gives a way to free users from complex hardware and software management by leaving such tasks to third parties, who create applications accessible to multiple users by a Web browser. Here, customers do not install anything on their premises. Also, they do not pay considerable upfront costs to purchase the software and the required licenses. They simply access the application Website, enter their credentials and billing details, and can instantly use the application that can be further customized for their requirements. The infrastructure maintains the specific details and characteristics of each customer's application and makes available when required on the provider side.

The SaaS model is useful for applications that can be adjusted to specific needs, with little further customization and serving a variety of users. This requirement characterizes SaaS as a one-to-many software delivery model, an application is shared across several users. This is the case of Customer Relationship Management (CRM) and Enterprise Resource Planning (ERP) applications that form general requirements for almost all the businesses. There will be similar requirements for the basic characteristics related to CRM and ERP in every enterprise, different requirements can be met with further customization. This scenario makes easy the development of software platforms offering a set of characteristics and supporting specialization and ease of integrations of new components. It constitutes the perfect candidate for hosted solutions, because the applications provided to the user are the same, and the applications itself give means to the users to shape itself on the basis of their requirements. Consequently, SaaS applications are naturally multi-tenant, which is a characteristic of SaaS. This characteristic enables providers to centralize and sustain the effort of managing large hardware infrastructures, optimizing resources by sharing the costs among the large user base, and maintaining and upgrading applications transparently to the users. Such costs constitute a minimal fraction of the usage fee paid for the software on the customer side.

The SaaS approach resides on top of the cloud computing stack. It fits into the cloud computing vision denoted by the acronym XaaS—everything as a service. Applications are provided as a service with SaaS. In the beginning, the SaaS model was useful only for lead users and early adopters. After cloud computing, there has been an increasing acceptance of SaaS as a feasible software delivery model. This results in the development of SaaS 2.0, which does not give a new technology but changes the manner in which SaaS is used.

Q.48. Classify the different types of clouds.

Or

Explain the types of cloud.

Ans. Clouds can be classified into following types –

(i) Public Clouds – Cloud computing environments that are open for public use alternatively for a large industry group. Some public clouds are Google, Amazon and IBM offerings.

(ii) Private Clouds – The cloud is implemented within the private premises of an institution and uses it to provide services to the users of the institution or a subset of them.

(iii) Hybrid (Heterogeneous) Clouds – A computing environment which combines multiple clouds where those clouds keep their unique identities, but are bound together as a unit. It identifies a private cloud that has been augmented with resources or services hosted in a public cloud.

(iv) Community Clouds – These clouds are particularly intended to address the requirements of a particular industry. The cloud is characterized by a multi-administrative domain encompassing various deployment models.

Q.49. Define the term Cloud IoT.

Ans. Cloud computing is a remote location technology that transformed the way of Information Technology. It provides Software as a Service (SaaS), Platform as a Service (PaaS) and Infrastructure as a Service (IaaS). Internet of Things (IoT) is radically changing the way of businesses operate and people interact with the physical world. Although things, Internet, and connectivity are the three core components of IoT, the value is in closing the gap between the physical and digital world in self-reinforcing and self-improving systems. The combination of cloud computing and Internet of Things (IoT) is known as Cloud IoT.

Amazon Web Services IoT, Google Cloud Platform, IBM Watson IoT, Microsoft Azure IoT Suite has provided the combination of Internet of Things (IoT) and Cloud computing. Cloud IoT enables to connect device to cloud services and other devices, secure data and interactions, process and act upon device data and enable applications to interact with devices even when they are offline and build a robust, maintainable, end-to-end Internet of Things (IoT) solution on cloud platform.
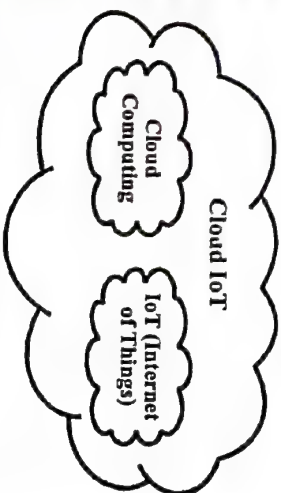

Fig. 5.23 Cloud IoT

### Q.50. Give advantages of using cloud for IoT.

Ans. As shown in fig. 5.24 there are multiple advantages to use cloud in IoT scenarios, as a result of which cloud connections have become a vital part of IoT deployments. The main advantage for IoT devices to be connected to the cloud is the always-on remote access to the data they collect (e.g. a temperature sensor) and the actions they can perform (e.g. turn on a light). Then there are other features, like allowing easy bootstrapping and management of devices and unlimited data storage. A centralized dashboard provides enhanced data visualization and analytics.
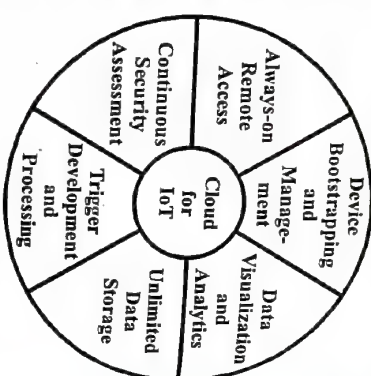
In regards to security, the cloud allows continuous assessment of security levels in IoT connections, identifying security risks (e.g. insecure authentication) and providing guidelines on how to mitigate them. For developers, there is one interesting feature of programming triggers for certain actions, using cloud services like AWS Lambda. For example let us imagine a trigger for movements sensed by a motion sensor. A movement event will set off the trigger which has programmed a set of actions, like sending an email to the owner of the sensor with the information about the movement and/or turning on the surrounding lights and an alarm sound. This can all be programmed through the AWS Lambda service and run continuously without any human intervention.

Fig. 5.24 Cloud for IoT

Q.51. Explain in brief about storage service in the cloud for IoT data management.

Ans. The reference architecture for the IoT data storage service in the cloud is shown in fig. 5.25.
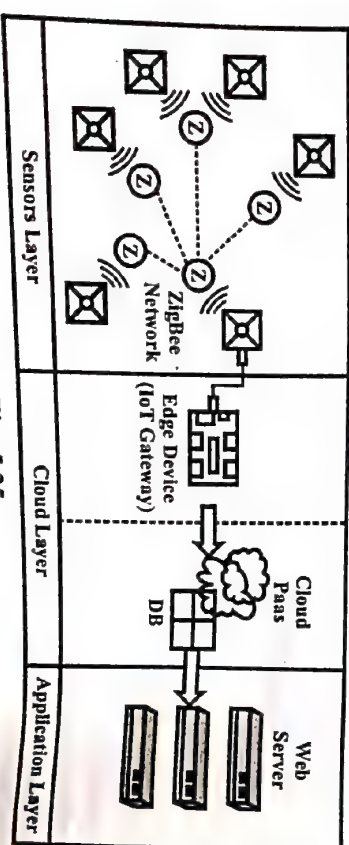
Fig. 5.25

The architecture has following three layers –

(i) Sensors Layer – It is responsible for collection and transmission of the data to the IoT gateway and it is also responsible for perceiving the environment. In our architecture implementation, at this layer, we exploit ZigBee networks. ZigBee networks permit standalone and scalar configuration and give a low-cost and low-power solution. However, as long the devices all agree upon how the data will be handed to the gateway, any other module or physical medium can be used. In specific, every ZigBee network can be configured with star topology in which the master node coordinates the process of transmitting slaves via requests. According to the request of masters, the slaves collect data from sensors.

(ii) Cloud Layer – It implements remote resource over the cloud data centers and in Internet. It involves an edge device. The functions of edge device is to convert data received from the sensor layer in data appropriate for the application layer. For data storage and retrieval over cloud data centers, the cloud platform as a service gives tools. In order to profit of the main merits of cloud computing in terms of elasticity and scalability. Data are stored in Azure DB.

(iii) Application Layer – It implements the business applications essential to manage and process AAL data. At the application layer, we exploit a web server that processes requests for data retrieval and storage at this stage of our work. It is proposed a storage mechanism based on NoSQL operating architecture, available on Azure known as Azure tables to store and give fast retrieval of information produced by the several devices attached to the cloud. NoSQL data bases have as one of it most notable features the non-relational data schema unlike storage systems based on relational paradigm, often permitting more flexible data storage. The Azure tables permit different number of attributes (columns) with the creation of key pair tuples (records). Hence, the Azure tables is able to offer flexible and low-cost storage and efficient searches.

Q.52. Write short note on IoT cloud based services. (R.G.P.V., May 2018)

Ans. The user act as an application service for the Internet of things. It obtains responses or feeds from the service or the application. The service which is provided by the IoT cloud for object calculations, data collection, data points and message passing. The service is the provision for generation and communication of alerts, triggers and feeds to the user. The cloud services are used to communicate the nodes with the server. A new data is generated at each time when the new value is recorded. Feed means a set of data points or objects or data streams which generate and communicate after

application of rules for filter, fusion, compression, analysis, aggregation, compaction and calculation. The feed may also considered for the alerts on the programmed triggers or alarms. It is said to be data stream generating and communicating at the present time intervals for an application or service use the push/subscribe or any other mode.

Q.53. Explain how to data store in IoT.

Describe data storage in IoT.

Or

Explain data storage in IoT ?   *(R.G.P.V., May 2018)*

Or

(R.G.P.V., May 2019)

Ans. Depending on the IoT infrastructure, devices can use different data storage and transmission mechanisms. There are IoT tools that store information received from sensors directly in their internal built-in-memory. The latter ones above all work autonomously and accumulate only necessary amount of information to perform real-time activities or to execute preset conditions with the help of aggregated data. The internal memory of these tools is usually very limited and it is meant only for sensor originated data.

Now-a-days a centralized data storage standard is used more extensively as shown in fig. 5.26. It allows the IoT devices to transmit data to a centralized server where it can be stored, analysed or managed. In theory, the number of connected devices and stored data can be infinite within such a system.
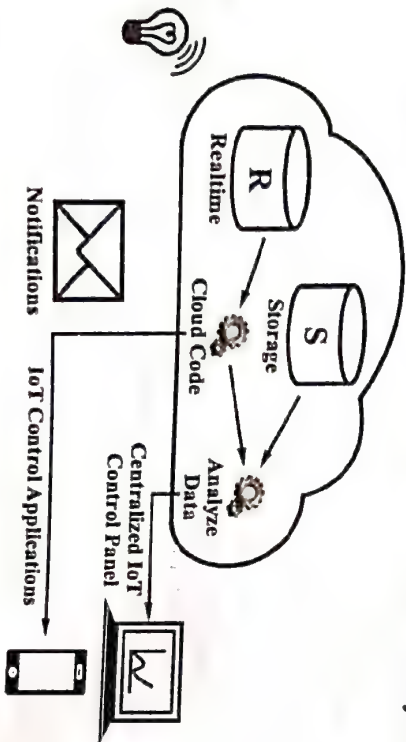

Fig. 5.26 Structure of Centralized Data Computing and Storage

Organizations increasingly require to store data produced from the IoT. This means deciding between cloud and on-premises IoT data storage is more important than ever. Internet of things defines multiple devices and sensors connected to the Internet, and their ability to capture and send data. Organizations use IoT devices to make improve customer satisfaction and good business

decisions. But after this data has been transferred, it requires to be amassed in storage system. This is the reason for companies to rethink their data storage infrastructures. While the cloud is look like a obvious choice for IoT data storage, but many organizations keep this information on site, believing it is either too sensitive or too expensive to store data in cloud. If you store your data on cloud, there are following advantages of cloud over on-premises –

(i) There is typically a more direct attachment between the device and the public cloud provider. This direct link means data can be stored off device faster, resulting in less storage on the device and per device cost is lower.

(ii) Now, the storage management is the cloud service provider's problem. The provider's job is to give a service and the organization just requires to use that services.

(iii) If the organization uses cloud compute to process IoT data, then the cloud makes an ideal storage location for that data.

But there are several issues of using the cloud for IoT data storage. The first one is security. In many conditions, the data being captured is legally sensitive or proprietary to the organization. The chance of that data being accessed in appropriately due to a security breach is a legitimate concern, but the reality is that most cloud service providers have excellent security or at least give the organization with the tools they need to secure their data. In most cases, breaches are caused by human errors, not by a weakness in provider security. Second is more pressing – the cost to store the IoT data set. While cloud storage costs are impressive when considered on a per-gigabyte, per-month basis, the allure wears off when the math is done to compute storing petabytes of information for decades. There are several attributes of cloud storage providers to consider beyond the price per GB per month when the decision is made to store data in the cloud.

Overall cost = Total capacity × No. of years to maintain

The cost to transfer data in and out of the cloud provider. Most providers do not charge for data as comes into the cloud, but several charge a bandwidth fee as it leaves. If the compute will be done on-premises, the transfer consideration is especially important, which means some transfers will occur no matter what.

On-premises IoT Data Storage – If IoT data will store on site, the next decision to make is whether to use a traditional NAS array or a private cloud architecture. When we shall use NAS array, the advantage will be familiarity, but often lacks cost-effective scaling and modern protocol support like objects and Amazon S3. The primary objective of an on-premises store is to ensure

data is stored cost-effectively but reliably and with minimal staffing. For long-term storage of large data sets, the on-premises strategy should be less expensive, the organization needs to be prepared to make that commitment, but because it will clearly become an IT function. The main purpose of most devices that fall under the IoT umbrella is to capture data. That means IoT devices will affect the storage decisions. You should start the infrastructure now. If you want to store data on site, IoT data is increasing fast. The rate of growth, as well as the time that organization want to save the data, will only accelerate.
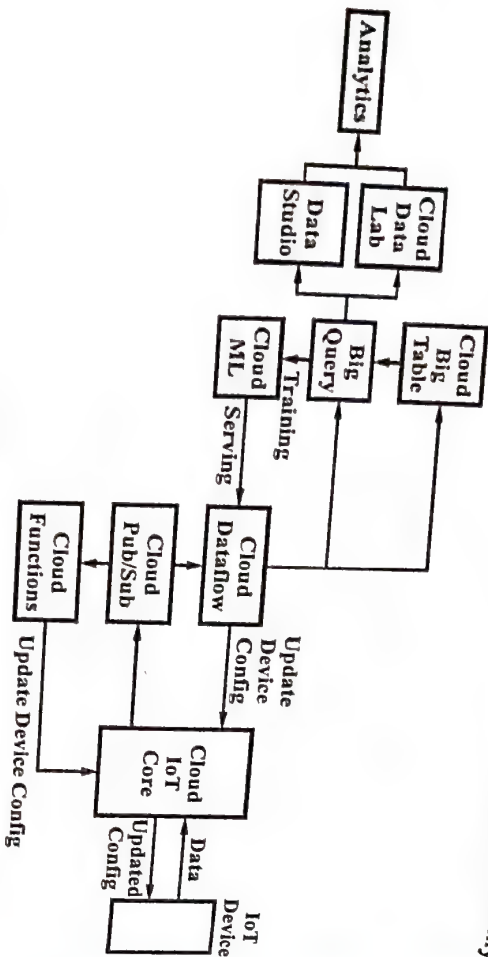

Fig. 5.27

Q.54. Name various cloud storage providers. Also write down merits and demerits of cloud storage.

Ans. There are various cloud storage providers. The name of top eight cloud storage providers are –

(i) Dropbox
(ii) Google Drive
(iii) Mega
(iv) OneDrive
(v) iCloud
(vi) Box
(vii) NextCloud
(viii) SpiderOak

Merits –

(i) Data is stored on the web in cloud.
(ii) Balance server loads by moving data among various hosting sites.
(iii) Cloud storage provides protection of data from disaster.
(iv) It keeps backup of stored data in cloud.
(v) Leveraging standardization, automation, virtualization helps to reduce expense.

Demerits –

(i) Some cloud storage providers provides limited bandwidth.
(ii) Data cannot be accessed without internet connection.
(iii) User cannot control security of his data in cloud.
(iv) User should copy and paste instead of drag/drop of his data.
(v) If user uses drag and drop option to move his data in cloud storage folder the data will be erased from his personal machine.

user want to copy his data on cloud storage folder. If user want to copy his data will be erased from his personal machine.

Q.55. Discuss about the cloud storage models and communication APIs.

Ans. **Cloud Storage Models –** As shown in fig. 5.28 cloud storage models are Amazon web services (AWS IoT), Google cloud platform, Microsoft Azure, and IBM Wastson etc.

Fig. 5.28 Cloud IoT Storage Models

AWS IoT is a managed cloud platform that allows connected devices to easily and securely interact with cloud applications and other devices. AWS IoT can support billions of devices and trillions of messages, and can process and route those messages to AWS endpoints and to other devices reliably and securely. With AWS IoT, client's applications can keep track of and communicate with all the devices, all the time, even when they are not connected.

Google Cloud Platform allows to build a robust, maintainable, end-to-end Internet of Things (IoT) solution on cloud platform. It creates streams of insight by extending client's infrastructure to the physical world.

Azure IoT suite brings the Internet of Things to life. We can connect our devices, analyze previously-untapped data, and integrate business systems and transform our company to uncover new business models and revenue streams.

IBM Watson IoT provides SDKs in C, C#, Mbed C++, Embedded C, Java, Python, Node.js and Node-Red. MQTT is the only supported protocol for data communications and can be used over Websockets. A secure HTTP(S) REST API is available for communication with other services and apps.

Communication APIs – Refer to Q.8 (iii) Unit-1.

Q.56. Explain in detail about Google Cloud platform.

Ans. Developers can code, test and deploy their applications with highly scalable and reliable infrastructure that is provided by Google and Google itself uses it. Developers have to just pay attention to the code and Google handles issues regarding infrastructure, computing power and data storage facility. Google Cloud is one of the popular IoT platform because of – Fast global network, Google's BigData tool. Pay as you use strategy, Support of various available services of cloud like RiptideIO, BigQuery, Firebase, PubSub, Telit Wireless solutions, Connecting Arduino and Firebase and Cassandra on Google cloud platform and many more.

Fig. 5.29 shows real time stream processing by Google. Devices send their status information to App Engine. So first load balancer makes sure that the load is balanced among various app engines. Then compute engine performs data computation and publication. Multiple instances of compute engine are available to insure reliability and scalability. The data is stored and backed up using cloud storage. Big query allows speedy insertion of data in tables of cloud database. The results can be presented to the end users by means of various analysis and visualisation technique.
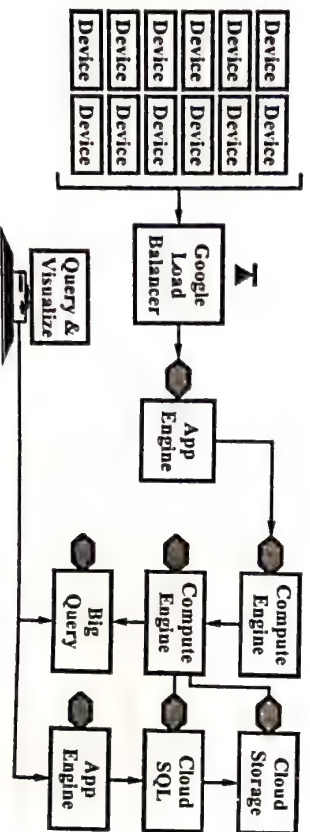

Fig. 5.29 Real Time Stream Processing Google IoT

Q.57. Explain various characteristics of Google Cloud platform.

Ans. Important characteristics of Google Cloud platform are as follows –

(i) Streaming Insights – Events of interest fire off continuously in the physical world, and data that is required for decision making cannot always wait for offline analysis. Internet-equipped sensors on any physical item

imaginable make it possible to ingest data continuously into the cloud, directly from the source at massive scale.

(ii) Tap into the World – A new type of device technology along with ubiquitous networking makes it easy and economical to mine information from any physical item and place. This untapped pool of data gives organizations visibility into parts of their operations previously considered "offline". Combined with real-time processing and management practices by enabling proactive resolution changes monitoring and management practices, and ultimately, predictive capabilities in response to real-time events.

(iii) From Small to Big (Data) – Each sensor-equipped device may be small and yields only incremental insight. Multiply this by hundreds, thousands, or millions of sensors all ingesting data to the cloud and the collective stream presents as a big data problem. Cloud Pub/Sub makes real-time, reliable processing of IoT data easy, and cloud storage products persist all big data efficiently and economically. IoT on cloud platform allows us to make extremely fast queries into any business and operating environment, without managing any infrastructure.

(iv) Global Fiber Network – Google operates its own private fiber network that spans the globe with over 70 points of presence across 33 countries, ensuring data to and from your devices gets delivered at ultra-low latency. Reliability and security are enhanced because IoT data do not have to travel the public Internet through the majority of its time in transit. Google's global network ensures that millions of devices and sensors distributed worldwide can deliver raw data efficiently so an organization can tap operational insight continuously with no disruption.

(v) Google-Grade Security – Whether device-to-cloud or cloud-to-device, security is the most important concern as IoT is increasingly used to support business-critical operations. All cloud platform APIs are secure by default with full encryption, backed by integrated and pervasive security across the entire infrastructure. Cloud IAM can ensure devices have access only to resources we explicitly designate.

Q.58. Explain in brief about the IBM BlueMix IoT platform.

Ans. IBM BlueMix is a platform as a service (PaaS) cloud which is developed by IBM. It supports programming languages like java, php, Python, Node.js, and many more. Integrated DevOps allows to build, run, deploy as well as to manage applications over IBM BlueMix cloud. BlueMix platform is based on Cloud Foundry open technology. It runs on Soft-Layer infrastructure. The IBM Internet of Things Foundation is powered by IBM's following leading

products and services – IBM DataPower Gateway, IBM WebSphere Application Server Liberty Core, IBM Informix TimeSeries, IBM MessageSight, Cloudant and SoftLayer.

IBM BlueMix platform provides powerful application access to IoT data visualization dashboard, and mobile IoT applications. We can create our IoT application with IBM BlueMix and then IBM provide REST and secure API to connect our device data with our application. IBM IoT foundation is the hub where we can set up and manage our connected devices. IBM IoT foundation uses MQTT protocol to securely transfer device data to cloud.

Q.59. What are the characteristics of IBMBlueMix ?

Ans. The characteristics of IBMBlueMix are as follows –

(i) Scale Institutional Expertise –

(a) Work smarter by giving any employee access to the company's collective knowledge.

(b) Infuse equipment and devices with deep domain knowledge and sensory input such as sound, images, videos and text.

(c) Give equipment and devices the power to reason and lean, and the ability to interact naturally with people.

(ii) Drive Environmental Leadership –

(a) Join the movement to use IoT for green initiatives that benefit the bottom line.

(b) Reduce energy and water consumption, eliminate waste across the supply chain, and optimize asset utilization.

(c) Cut costs while reducing CO_2 emissions and enhancing your brand image.

(iii) Lead Industry Transformation –

(a) Find new ways to monetize value through asset-based online marketplaces.

(b) Garner new revenue from existing products and services.

(c) Partner in new ways across industries.

(d) Disrupt competitors by seizing new IoT opportunities first.

(iv) Boost Operational Performance –

(a) Gain a whole new level of visibility across the extended supply chain.

(b) Optimize performance, empower employees and lower costs.

(c) Monitor assets and equipment to enable predictive maintenance.

(d) Increase throughput and optimize resource use.

(e) Run more energy and cost-efficient facilities.

(v) Enhance the Customer Experience –

(a) Offer products and services that continuously adapt to the connected customer.

(b) Gain valuable insights throughout the product life cycle that inform product development.

(c) Achieve a new level of customer engagement to boost loyalty, drive revenue and create differentiation.

(d) Speed up development and accelerate time to market.

Q.60. Discuss about the Microsoft IoT platform.

Ans. According to Microsoft, Internet of Things for a business starts with the things that are involved in the business and affect it the most. It starts from the infrastructure of the business by adding devices and services into it with technological expansion, allows the business to get insight of the information and make powerful and informative business decision. Thus, here IoT is actually Internet of our things. The piece of data about customers, sales, business processes or other inventory data is valuable asset to an organisation and it can help to power the business. Microsoft has the ability to transform a current business to a flexible intelligent system with existing investment to achieve true flexibility. Microsoft provides Microsoft Azure Intelligent System Service which forms an integrated platform and services that builds Internet of Things systems and applications by gathering, storing and processing data. Intelligent systems services build upon Microsoft Azure helps organizations to securely connect, manage, capture and transform machine generated data into valuable information. Power BI, Office 365 and HD insight are powerful Microsoft assets that can be used to produce meaningful insights. Intelligent Systems Service provides agents and open-source agent software which help to support heterogeneous operating systems and protocols for Internet of Things System. With the help of Microsoft Cloud Compute facility, scalable data collection, processing and analysis can be done for a business processes. Cloud provides solution for data storage, data processing, data consumption and data analysis on real time or latent data.

The core of Microsoft IoT foundation is Microsoft Azure cloud platform. It provides connectivity to millions of devices and sensors with IoT application. It provides remote access, monitoring, and content distribution and configuration management facilities for connected devices. It provides big data analysis, social as well as business integration and dash boarding tools to IoT application to build an IoT solution.

Q.61. Discuss about the AWS IoT platform.

Ans. Amazon Web Services (AWS) allows Internet of Things on a global scale by facilitating security, services and support. It facilitates immediate access to desired computing power by means of Amazon Elastic Cloud Compute (EC2). It helps to performs big data analytics and supports high volume data. Amazon Kinesis helps to ingest data from thousands of sensors. AWS provides security to our data which can be in transit or at rest. AWS provide pay-as-you-go model for IoT applications. It has multiple pricing models like tiered pricing, Reserved Instances, and an active marketplace. AWS supports on demand pricing.

It provides access to more storage, compute capability and global resources when needed. AWS provides flexibility for IoT applications in terms of tools, programming languages, data management and other infrastructure resources. ActiveMq and Mosquitto servers help in managing and analysing IoT applications. User identity, device analytics, and device messaging/notifications are all common parts of an Internet of Things application. Amazon Web Services provides services that take the effort out of these important parts of an application. Services like Amazon Cognito, Amazon Mobile Analytics, and Mobile Push to take care of the undifferentiated heavy lifting while we focus on the differentiated benefits of application. As shown in fig. 5.30 Amazon Kinesis can collect high throughput data from devices and gateways, and then it can analyze and store it over cloud so that applications can consume it and can generate quick decision. It can support data up to any scale.

Fig. 5.30 Data Streaming from Gateway to Cloud by Amazon Kinesis

Q.62. What are the characteristics of AWS IoT platform ?

Ans. The characteristics of AWS IoT platform are as follows –

(i) AWS IoT Devices SDK – AWS IoT provides an SDK to help us easily and quickly connect hardware device or mobile application. The AWS IoT device SDK enables our devices to connect, authenticate, and exchange messages with AWS IoT using the MQTT, HTTP, or WebSockets protocols. The device SDK supports C, JavaScript, and Arduino, and includes the client libraries, the developer guide, and the porting guide for manufacturers.

(ii) Device Gateway – The AWS IoT device gateway enables devices to securely and efficiently communicate with AWS IoT. The device gateway

can exchange messages using a publication/subscription model, which enables one-to-one and one-to-many communications. With this one-to-many communication pattern AWS IoT makes it possible for a connected device to broadcast data to multiple subscribers for a given topic. The device gateway supports MQTT, WebSockets, and HTTP 1.1 protocols and can easily implement support for proprietary or legacy protocols. The device gateway scales automatically to support over a billion devices without provisioning of infrastructure.

(iii) Rules Engine – The Rules Engine makes it possible to build IoT applications that gather, process, analyze and act on data generated by connected devices at global scale without having to manage any infrastructure. The Rules Engine evaluates inbound messages published into AWS IoT and transforms and delivers them to another device or a cloud service, based on business rules we define. A rule can apply to data from one or many devices, and it can take one or many actions in parallel. The Rules Engine can also route messages to AWS endpoints including AWS Lambda, Amazon Kinesis, Amazon S3, Amazon Machine Learning, Amazon DynamoDB, Amazon CloudWatch, and Amazon Elasticsearch Service with built-in Kibana integration.

(iv) Registry – The Registry establishes an identity for devices and tracks metadata such as the devices attributes and capabilities. The Registry assigns a unique identity to each device that is consistently formatted regardless of the type of device or how it connects. It also supports metadata that describes the capabilities of a device, for example whether a sensor reports temperature, and if the data are Fahrenheit or Celsius.

Q.63. What is the design principle for web connectivity ?

Ans. An IoT or M2M device network gateway is required to connect with the web servers for their functioning. A web connectivity is enabled with a communication gateway, on the other side web connectivity for network of connected devices can be enabled with IoT specific protocols or techniques. IoT device data accumulation is enabled by the server. This data can be used by application, collaboration, service and processes.

There are some key terms which need to be known to learn the design principle –

Application – It is a software for applications like calculating and transmitting the calculated data, generating and transmitting a SMS, obtaining a message from a particular sender etc.

Application Programming Interface – API receives message from one end, process data and send output to other end. It is a software components which enable easier development of an application. API defines the methods

for a developer, which adds the building blocks together to develop an application. Inputs, outputs and underlying types are involved in a building block.

Web Services – It is a software servicing. It employs web protocols, web objects or WebSockets. For example, traffic density reports, weather reports service, street-lights monitoring and controlling service.

Objects – Collection of resources is known as objects just like collection of data and functions to operate on that data.

Object Instance – It can be one or more than one for an object. Just like, roll_no. A roll_no can multiple object instances like abc_roll_no, xyz_roll_no, pqr_roll_no, etc. Object instance is used for weather report object for reporting about the rains.

Object Model – Usage of objects for data transfer, values, messages, and creation of one or more object instances is known as "object model".

Class – Object-oriented programming languages like Java and C++ use concept of class which generates one or more object instances.

Communication Gateway – It works like communication protocol translator for provisioning communication capabilities. Just like, gateway can communicate between IP network and ZigBee.

Client – It is a software object, which builts request for data, resources. Client can consist of one or more object instances. It can consist of one or more APIs for performing communication to the server.

Server – It is a software, which transmits a response on a client request.

Proxy – It is an application software which obtains requests from the client for the responses retrieved and which also obtains a response from the server for usage of a client or application.

Communication Protocol – It can be referred to the rules and conventions for communication between systems and networked devices. It also includes formatting rules for how data is packaged for sending and receiving messages, and header, its field and their meanings. In addition, it comprises mechanisms for devices or systems to recognize and make connections with each other.

Firewall – It helps to protect the server from unauthentic resources.

Path – Path specification is URI or URL type. When accessing a resource, path is a navigation path between two ends.

Universal Resource Identifier – It uses for stored data like address book and the structure of URI is given below – /Contacts/First_Character_D

for a resource set directory contacts containing resource repository first_character_D for contacts by first character D and resources providing information about a contact.

Universal Resource Locator – It used for retrieving a resource from a client site. These saved resources are accessed through Internet protocols at a document and at a remote server. For example, http://www.google.com/ for search data.

Browser – It is a display software, which is used on client machine. It displays hypertext that enables navigation to the hypertext links represented on the client screen, and it displays graphical user interface of the software, display form and display server responses.

WebSocket – It is an API for bidirectional communication. It has lower latency than HTTP method of unidirectional communication and it has much less header size compared to HTTP communication.

Framework – It is used for time saving and good programming and low bugs. It has predefine libraries, which is used by programmer to write a program. It has provisions for a multiple libraries, and APIs comprising those that can be selectively changed by user code in applications.

Q.64. Explain the following communication environments –

    (i) Unconstrained environment
    (ii) Constrained RESTful environment.

Ans. (i) Unconstrained Environment – Hyper text transfer protocol and RESTful hyper text transfer protocol are used by web applications for web server communication on web client. For the Internet data routes over IP networks. A web object have 1000 of bytes. IP and TCP protocols are used by web services and applications for transport layers and Internet network.

    (ii) Constrained RESTful Environment – In a LAN machine-to-machine or IoT devices communicate between themselves. A device typically obtains or transmits 10s of bytes. The data collected after consolidating and enriching from many devices which have 100s of bytes. A gateway in the communication framework enables the data of networked devices which communicate over the Internet using the REST software architecture.

Q.65. Explain in brief about simple object access protocol (SOAP).

Ans. SOAP is an open source protocol. It has been approved by W3C. With the help of XML, SOAP protocol is used for exchange of objects between applications. Besides, it is a protocol for access web services. It is extensible and can also be employed for APIs for service oriented architecture (SOA)

and the web services. Its usage is not dependent of the application language and OS platform. The formats and method of communicating the messages are specified by the SOAP.

SOAP can be used to enable development of application and APIs. In the case of Internet application development, Microsoft's .NET architecture helps SOAP. With the help of HTTP and XML, SOAP is used to connect the GUI applications to web servers.

SOAP v1.2 second edition specifications are recommended through W3C. In this specifications, part 0 represents a primer, part 1 represents a messaging framework and part 2 represents adjuncts. SOAP v1.2 specifications provision for XML-binary optimized packaging, resource representation SOAP header block, assertion and test collection, and SOAP message transmission optimization mechanism.

After the specifications, a body element is used with the SOAP in an envelope. The body element contains the SOAP message intended in case of ultimate message end points.

The HTTP POST or HTTP GET are the SOAP request. At least two HTTP headers are specified by the HTTP POST, i.e., content length and content type. After the HTTP binding with the SOAP, a SOAP method employs HTTP request/response. The request and response complies by the rules of SOAP encoding.

Q.66. Explain in brief about REST architectural style of designing software components.

Or

List features of REST architectural style of designing software components. *(R.G.P.V., May 2018)*

Ans. The full form of REST is "Representational State Transfer". REST architecture style was introduced by W3C technical architecture group. The group worked in parallel with the help of HTTP 1.1. REST represents a coordinated set of constraints which are employed in a distributed hypermedia during design of software parts. Design of software components depends on the features of stateless, client-server, catchable communication with the help of a protocol. REST practices and constraints are used by Internet. REST represents a easier alternative in case of web services description language and SOAP the REST is a easy option. SOAP is increasingly replace by REST style web resources and resource-oriented architecture. Applying special interaction constraints to data elements, components, connectors and objects for realising the architectural properties of REST. Separation of concern is an architectural feature of REST, separation of concern means, for the client,

data storage at server is not a concern, and client equipment may port on other objects. User interface and user state are not concern to the server, it is a concern of client. The component implementation, increases the scalability of pure server components, decreases connector semantics difficulties and increasing of performance tuning effectiveness are easiest by architectural properties like separation of concerns of client-server of REST. Separation of resources from representation conceptually done themselves. Here representation resources are which returned to the client. For the use of client-server mode of communication and layered system architectural approach, REST software architecture style are self explained. Performance and creation of scalable web objects and services are features of client-server interactions. Ability to help lot of components and lot of interactions among components is known as scalability. The clients that are connected via an intermediate layer is called layered system. By constraining the messages the REST allows intermediate layer processing such that the interactions at every layer are self explained. Generally, a client cannot be made aware whether it is associated directly to end server. At two ends, in transcoding and enabling usages of various protocols intermediate layers may render assistance. When using greater scales and shared caches performance may improve by use of intermediate system. Dividation of caches between two layers is known as shared caches. Application state can be shown by links which may be employed for a future interaction if the user chooses those links and begans a new state transition.

Q.67. Write a short note on RESTful.

Ans. All interactions involved in the applications when fully conform to the REST constraints then these interactions are known as RESTful. RESTful APIs obey with these constraints and hence conform to REST architectural style. With RESTful APIs web services add to the representational state transfer architectural constraints. For HTTP access, architectural style of REST can be used by GET, POST, PUT and DELETE methods for building web services and resources.

Q.68. Explain in brief about RESTful HTTP APIs.

Ans. GET, PUT, POST and DELETE are the standard methods of HTTP. RESTful which is dependent on HTTP. Following APIs are used by RESTful which is dependent on HTTP –

(i) URLs/URIs examples are such as http://weatherMsgService.com /weatherMsg/ and other hypertext links.

(ii) Generally, REST-based web objects communicate, over the HTTP. To the REST architectural style WWW itself shows the biggest implementation

of a system conforming. Communication over the HTTP is RESTful HTTP system feature and employ commands (i.e. PUT, DELETE, POST GET) same as in HTTP.

Q.69. What do you understand by RESTful HTTP verbs ?

Ans. Generally, resource repositories with identifiers are used by REST interfaces. The standard verbs which can be operated in temperature app, device network are given below –

(i) DELETE command retrieves web objects from client and transmits data to remote servers.

(ii) PUT command replaces the resource item of the repository or also replaces the whole resource repository with another resource repository.

(iii) To get a list of the URIs for resource repository of the resources and other information of the members in the repository GET command can be used. GET gives a representation of the resource item of the repository. Representation is described in an suitable Internet media type.

(iv) A new entry, for the resources, in the resource repository is created by POST command. Generally, the entry's of new URI is done automatically and is returned by the operation. The resource item is considered as a repository on its own right by an option which is not generally used and generate a new entry list in it.

Q.70. What do you understand by WebSocket ? Explain.

Or

Write short note on WebSockets.

(R.G.P.V., May 2019)

Ans. Web protocol's specifications are described by RFC 6455. WebSocket is a protocol which is recommended by IETF.

Over the same connection many applications and instant messaging require bidirectional data exchanges. WebSocket allows bidirectional communication over a single TCP connections.

Opcode and other fields at a WebSocket frame is shown in fig. 5.31 (c). WebSocket API provisions for functions, events and attributes are shown in fig. 5.31 (b). Occurrence of new situation is known as event, as soon as listening occurs an event handling function executes. The handling function is also called callback action. Fig. 5.31 (a) depicts data bidirectional communication using WebSocket APIs among servers and browsers. Fig. 5.31 (a) also shows the bidirectional communication between web objects by using WebSocket.
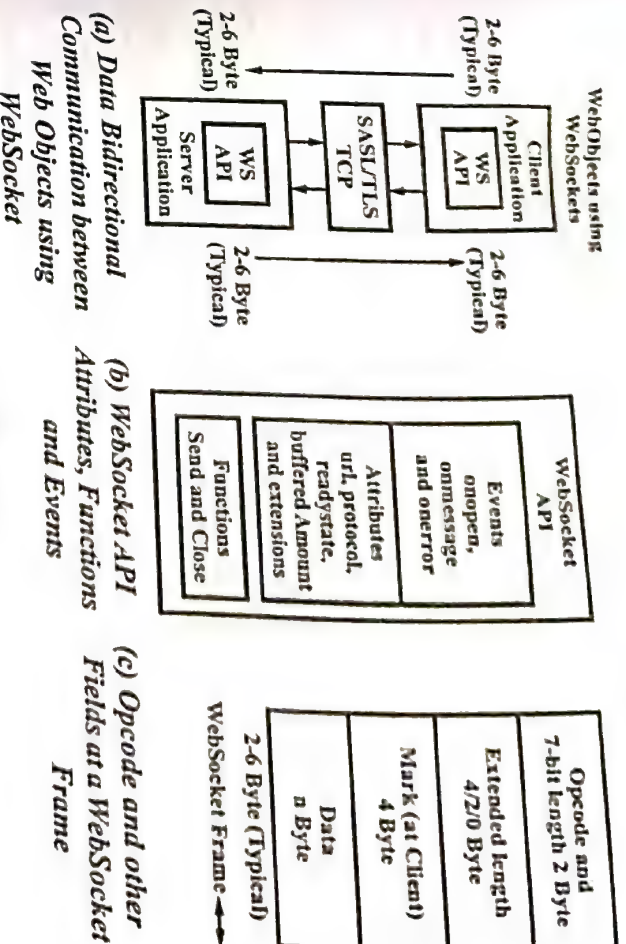

WebObjects using WebSockets

Fig. 5.31

(a) Data Bidirectional Communication between Web Objects using WebSocket

(b) WebSocket API Attributes, Functions and Events

(c) Opcode and other Fields at a WebSocket Frame

Supersedes the existing bidirectional communication approaches which use HTTP at application layer is used by WebSocket-protocol-based design. Url, protocol, readystate, buffered amount are WSAPI attributes which are extendable. Send and close are the functions of API. A frame has header plus data. WebSocket frame has 2 byte header which can be extended to, 0, 2 or 4 byte plus data bytes. For example, 4 byte mark is added by client side to identify a frame. The WebSockets allow simple usage of existing authentication, infrastructure, filtering and proxies. The protocol is supported by multiple browsers. Due to no-wait property of WebSocket HTTP case polling at periodic intervals are not needed.

Q.71. What are the features of WebSocket ?

Ans. Following are the features of WebSocket –

(i) Small header size.

(ii) No new latency period because no new connection that will require new header.

(iii) WSAPIs facilitate live content and the generation of real-time games.

(iv) Protocol is an independent TCP-based protocol. Handshake is interpreted by HTTP servers like an upgrade request this is the relationship of protocol to HTTP.

(v) For regular WebSocket connections protocol use port 80 when using ws:// and for WebSocket connections tunnelled over transport layer security protocol use port 443 when using wss://.

(vi) Protocol is trying to be compatible with HTTP-based server-side software and intermediaries, so that both HTTP clients use single port for talking to that server and WebSocket clients talking to that server. Hence, handshake of WebSocket client's is an HTTP upgrade request. Response from the server is also as an HTTP upgrade.

(vii) Protocol defines six frame types and leaves ten reserved for future use.

(viii) WebSocket with the support from intelligent and business analyst applications and processing through web server or XMPP server and gateway for connecting the device network with the IP network.

(ix) After successful handshake, clients and servers exchange the information.

(x) Extensibility of cloud services to super chat and information theory querying chat extensibility to client-server architecture.

(xi) It is not necessary that a WebSocket message correspond to a particular network layer framing. Frame has an associated type. Every frame related to the same message involves the same type of data.

ATTACKS IN IoT SYSTEM, VULNERABILITY ANALYSIS IN IoT, IoT CASE STUDIES, SMART HOME, SMART FARMING, ETC.

Q. 72. What are the different types of attacks in IoT system ?

Ans. The different types of attacks in IoT system are as follows –

(i) Denial-of-service (DoS) – This kind of attack is an attempt to make a machine or network resource unavailable to its intended users. Due to low memory capabilities and limited computation resources, the majority of devices in IoT are vulnerable to resource enervation attacks.

(ii) Physical Attacks – This sort of attack tampers with hardware components. Due to the unattended and distributed nature of the IoT, most devices typically operate in outdoor environments, which are highly susceptible to physical attacks.

(iii) Reconnaissance Attacks – This refers to unauthorized discovery and mapping of systems, services, or vulnerabilities. Examples of this type of attacks are scanning network ports, packet sniffers, traffic analysis, and sending queries about IP address information.

(iv) Cyber-crimes – The Internet and smart objects are used to exploit users and data for materialistic gain, such as intellectual property theft, identity theft, brand theft, and fraud.

(v) Destructive Attacks – Space is used to create large-scale disruption and destruction of life and property. Examples of this type of attacks are terrorist and revenge attacks.

(vi) Access Attacks – In this kind of attack, unauthorized persons gain access to networks or devices to which they have no right to access. There are two different types of access attack, the first is physical access, whereby the intruder can gain access to a physical device. The second is remote access, which is done to IP-connected devices.

(vii) Supervisory Control and Data Acquisition (SCADA) Attacks – As any other TCP/IP systems, the SCADA system is vulnerable to many cyber attacks. The system can be attacked in any of the following ways –

(a) Using denial-of-service to shut down the system.

(b) Using Trojans or viruses to take control of the system.

(viii) Attacks on Privacy – Privacy protection in IoT has become increasingly challenging due to large volumes of information easily available through remote access mechanisms. The most common attacks on user privacy are as follows –

(a) Data Mining – Enable attackers to discover information that is not anticipated in certain databases.

(b) Cyber Espionage–Using cracking techniques and malicious software to spy or obtain secret information of individuals, organizations or the government.

(c) Eavesdropping – Listening to a conversation between two parties.

(d) Tracking – A user's movements can be tracked by the devices unique identification number (UID). Tracking a users location facilitates identifying them in situations in which they wish to remain anonymous.

(e) Password-based Attacks – Attempts are made by intruders to duplicate a valid user password. This attempt can be made in two different ways –

(1) Dictionary Attack – Trying possible combinations of letters and numbers to guess user passwords.

(2) Brute Force Attacks – Using cracking tools to try all possible combinations of passwords to uncover valid passwords.

Q.73. Describe the primary security goals in IoT system.

Ans. The primary security goals in IoT system are as follows –

(i) Confidentiality – This is an important security feature in IoT, but it may not be mandatory in some scenarios where data is presented publicly. However, in most situations and scenarios sensitive data must not be disclosed or read by unauthorized entities. For instance patient data, private business data, and/or military data as well as security credentials and secret keys, must be hidden from unauthorized entities.

(ii) Auditing – A security audit is a systematic evaluation of the security of a device or service by measuring how well it conforms to a set of established criteria. Due to many bugs and vulnerabilities in most systems, security auditing plays an important role in determining any exploitable weaknesses that put the data at risk. In IoT, a systems need for auditing depends on the application and its value.

(iii) Integrity – To provide reliable services to IoT users, integrity is a mandatory security property in most cases. Different systems in IoT have various integrity requirements. For instance, a remote patient monitoring system will have high integrity checking against random errors due to information sensitivities. Loss or manipulation of data may occur due to communication, potentially causing loss of human lives.

(iv) Non-repudiation – The property of non-repudiation produces certain evidence in cases where the user or device cannot deny an action. Non-repudiation is not considered an important security property for most of IoT. It may be applicable in certain contexts, for instance, payment systems where users or providers cannot deny a payment action.

(v) Authentication and Authorization – Ubiquitous connectivity of the IoT aggravates the problem of authentication because of the nature of IoT environments, where possible communication would take place between device to device (M2M), human to device, and/or human to human. Different authentication requirements necessitate different solutions in different systems. Some solutions must be strong, for example authentication of bank cards or bank systems. On the other hand, most will have to be international, e.g., ePassport, while others have to be local. The authorization property allows only authorized entities (any authenticated entity) to perform certain operations in the network.

(vi) Availability – A user of a device (or the device itself) must be capable of accessing services anytime, whenever needed. Different hardware

and software components in IoT devices must be robust so as to provide services even in the presence of malicious entities or adverse situations. Various systems have different availability requirements. For instance, fire monitoring or healthcare monitoring systems would likely have higher availability requirements than roadside pollution sensors.

Q.74. Explain privacy goals in IoT system.

Ans. Privacy is an entity's right to determine the degree to which it will interact with its environment and to what extent the entity is willing to share information about itself with others. The main privacy goals in IoT are as follows –

(i) Privacy During Communication – It depends on the availability of a device, and device integrity and reliability. IoT devices should communicate only when there is a need, to derogate the disclosure of data privacy during communication.

(ii) Privacy in Devices – It depends on physical and commutation privacy. Sensitive information may be leaked out of the device in cases of device theft or loss and resilience to side channel attacks.

(iii) Privacy in Processing – It depends on device and communication integrity. Data should be disclosed to or retained from third parties without the knowledge of the data owner.

(iv) Identity Privacy – The identity of any device should only be discovered by authorized entity.

(v) Location Privacy – The geographical position of relevant device should only be discovered by authorized entity.

(vi) Privacy in Storage – To protect the privacy of data stored in devices, the following two things should be considered –

(a) Possible amounts of data needed should be stored in devices.

(b) Regulation must be extended to provide protection of user data after end-of-device life (deletion of the device data (Wipe) if the device is stolen, lost or not in use).

Q.75. Define the term vulnerability in IoT system.

Ans. The term vulnerabilities refers to weaknesses in a system or its design that allow an intruder to execute commands, access unauthorized data, and/or conduct denial-of-service attacks. They can be found in variety of areas in the IoT systems. In particular, they can be weaknesses in system

hardware or software, weaknesses in policies and procedures used in the systems and weaknesses of the system users themselves.

IoT systems are based on two main components which are, system hardware and system software, and both have design flaws quite often. Hardware vulnerabilities are very difficult to identify and also difficult to fix even if the vulnerabilities were identified due to hardware compatibility and interoperability and also the effort it take to be fixed. Software vulnerabilities can be found in operating systems, application software, and control software like communication protocols and devices drives. There are a number of factors that lead to software design flaws, including human factors and software complexity. Technical vulnerabilities usually happen due to human weaknesses. These are the results of not understanding the requirements comprise starting the project without a plan, poor communication between developers and users, a lack of resources, skills, and knowledge, and failing to manage and control the system.

Q.76. Draw and explain infrastructure for a WSN IoT application in smart home control and monitoring system.

Ans. In IoT-based applications and services, WSNs used as sub-system. WSN can functions as a source of data and connect the other systems to the Internet through a gateway and access point.

A smart home control and monitoring system use WSN for following applications –

(i) Mobile, laptop, WiFi and Internet, IP-TV, video conferencing, video-on-demand, video surveillance.

(ii) WSN and wireless actuator nodes for home security access control and security alerts, lighting control, fire detection, gas leak detection, temperature monitoring and HVAC control and automated meter reading.

These devices built using ZigBee IP are called ZigBee devices.

ZigBee IP is an enhancement of IPv6 connectivity, which uses RFD (Reduced Function Device). ZigBee devices from WPAN (Wireless Personal Area Network) in a smart home network. ZigBee protocol, which is a IEEE 802.15.4 standard protocol for physical/data-link layer can also be used by WSN nodes routing nodes and co-ordinating nodes.

A smart home network using cluster of ZigBee WSN and actuator nodes, routers and coordinators connected through gateway and set of RPL routers is shown in fig. 5.32. ZigBee devices receives data packets from IPv6 addresses and communicate with IoT and M2M IoT applications.
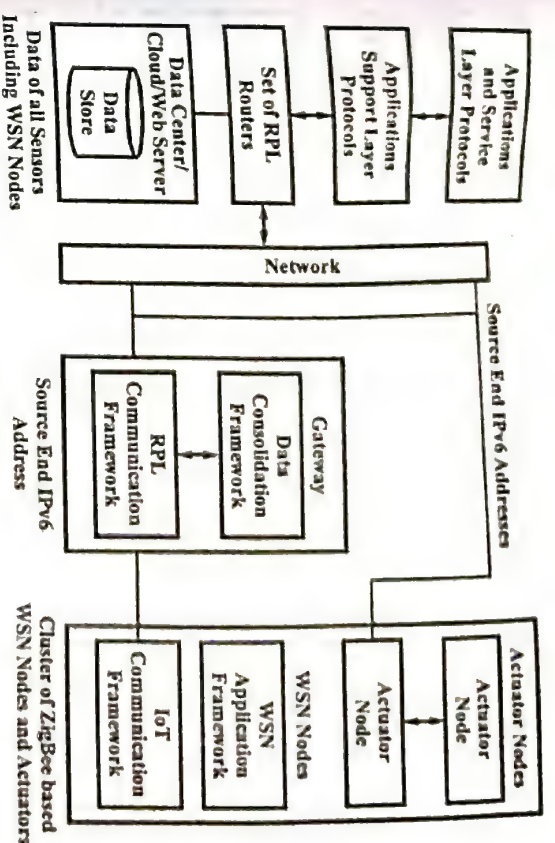
Fig. 5.32

Q.77. Explain IoT based smart city streetlights control and monitoring system.

(R.G.P.V., Nov. 2019)

Ans. A smart city streetlights are hosted with a lamppost, i.e., sensors and WSN actuator. The lights are turned on and off with the help of actuator. Messages are enabled with the help of sensors for lamp functioning status, traffic and ambient light.

The lights will be turned-on, when the ambient light will be above threshold level. The traffic density messages communicate to traffic signal controlling service. The WSN transceiver deploy sensors in case of detecting traffic presence and traffic density. By using these sensors, the lights are turned-off, when traffic is not present. This provides in energy saving.

A lamppost is an active node for service network. City services may deploy lampposts for streetlighting systems like information networks. Also, the WSN transceiver may receive data with other services like traffic signalling service, Wi-Fi service or security service and resend onto the network of WSNs and then to access points.

In real time, every transceiver can receive and resend at the lamppost. Events, alerts, messages, triggers and notifications from a number of services are used to send for services like traffic signalling air-quality index monitoring services, smart parking, emergency services and hospitals, security services for home, banks and important public places.

The following operations are performed by a control and monitoring service for city streetlights –

(i) Streetlights are measured and monitored, and traffic parameters are measured at preset intervals in real time.

(ii) For configuring and communicating within the WSN network, the program uploads each WSN.

(iii) CISCO IoT, IOX and Fog, my.openHAB, TCUO, Nimbits, AWS or Bluemix can be cloud platform from Watson analytics.

(iv) Platform for processes, analyses and visualisation of the data and database information are provided by cloud node.

(v) Analytics and AI for optimising, monitoring and control functions are obtained by cloud node.

(vi) Activates the alerts and triggers and integrates data.

(vii) At periodic intervals, runs at data-adaptation layer in faulty or inaccessible sensors.

(viii) The WSN network messages are communicated. A coordinator, which deploys the data adaption, store, time, location, IDs stamping and gateway interfaces, is connected by the network.

(ix) After aggregating, compacting and processing at data-adaptation layer, coordinator generates and communicates alerts, triggers, messages and data.

(x) A database which can be transferred to the cloud for processing and for cloud data store, is created and updated by coordinator in real time.

(xi) Messages to the access point are transmitted at the preset intervals. The access point is connected to the coordinator.

(xii) A control and monitoring service at the WSNs and gateways uploads the programs using the OTP properties. At the cloud node, an OTP module gives OTP management and uploads connectivity programs for gateways.

Q.78. Discuss any one case study of IoT in detail.(R.G.P.V., May 2019)

Ans. Refer to Q.77.

Q.79. Explain in brief about data flow diagram and domain architecture reference model for the WSNs networks of city streetlights central control and monitoring service.

Ans. Data flow diagram and domain architecture reference model for the streetlight monitoring service is shown in fig. 5.33.
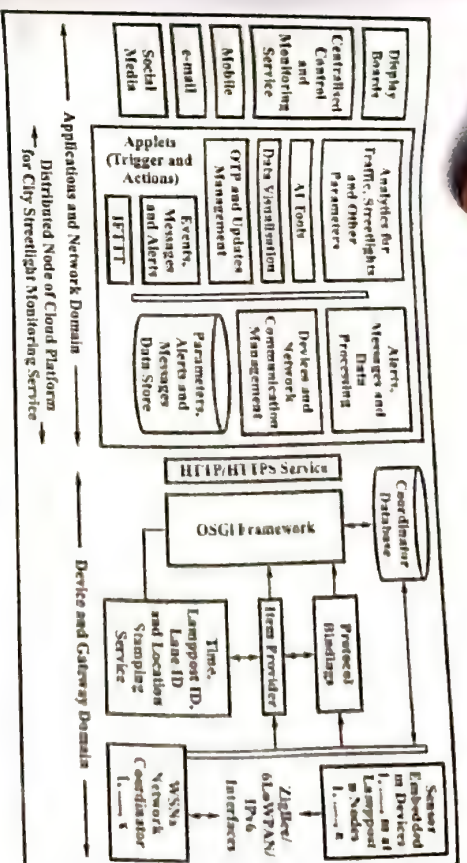
Fig. 5.33

Device and Gateway Domain – Hardware and software components and modules are given below –

Hardware – Hardware at a WSN controls, m embedded devices. With the help of ZigBee/6LoWPAN/IPv6 protocol, the n WSN node networks communicate with each other. x coordinators are deployed by smart city streetlight. With the help of LPWAN or ZigBee IP wireless interfaces, each network communicates with a coordinator. Data store, protocol binder, item provider and gateway are coordinator functions. A WSN is deployed by each lamppost. A set of sensor data is sensed by each node. Arduino boards with ZigBee or ZigBee IP boards can be used by sensor circuits. A network of ZigBee devices is formed by each WSN interfacing with other WSNs. Following parameters are calculated by a WSN –

(i) Lamppost condition, whether non-functional or not.

(ii) Density of traffic

(iii) Presence or absence of traffic in surroundings

(iv) Ambient light status, whether below or above a preset threshold level.

It is not necessary that each lamppost measures traffic parameters. Sensing devices are configured by each WSN so that a measurement deactivates or activates from the coordinator and central monitor service. At different preset intervals, configuring the node allows each parameter calculation. The actuator is configured by each WSN for allowing the lights on and off as per commands.

With the help of ZigBee, a group of WSNs communicate among themselves and form a network. With the help of LPWAN, each network access point receives the messages from each node. A gateway is associated by each access point. With the help of LPWAN, each gateway communicates to the cloud.

Software – At devices and gateway domain, open source IDE or Eclipse IoT stack which have OSGi can be used in software development. Sensor-IDs, lamppost-ID, lane-ID, subgroup-ID are assigned to each WSN. A subgroup of wireless sensor nodes create a WSN network and an assigned network ID. A coordinator-ID assigns to each coordinator. Each coordinator includes three modules –

    (i)   Protocol binding module

    (ii) For communication of queried items, alerts, messages and data, item provider module.

    (iii) Time, lamppost ID, lane ID and location stamping service.

For Java codes an open source OSGi framework can be used by coordinator. Streetlights, lanes and lane subgroup data are stored in database which is available at the coordinator.

Applications and Network Domain – HTTP or HTTPs service is used for internet connectivity. The IP protocol network routers connect each coordinator with a distributed node. Multiple distributed nodes are deployed by cloud platform for city streetlight monitoring service. The distributed node platform –

    (i)   Provides the alerts, messages, and data processing module

    (ii) Provides the devices network and communication management module

    (iii) Provides the AI tools.

    (iv) Provides the event messages, triggers and alerts for central control and monitoring services.

    (v) Provides the analytics tools for traffic, streetlights and other parameters.

    (vi) Provides the data store for parameters, alerts and messages.

    (vii) Provides the data visualization tools.

    (viii) Provides the coordinators, networks and nodes update management with the help of OTP.

    (ix) IFTTT for communication to mobile, e-mail, social media like instagram, musically, twitter, etc. and web services and applications.

Q.80. Write a code for streetlights control and monitoring in Java.

Ans.

```
Class Main

package com.main.execution.files;

import java.sql.Time;
import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.HashMap; // JSON object is subclass of java.util.HashMap
import org.json.simple.JSONObject;
import org.hibernate.Session;
import org.hibernate.SessionFactory;
import org.hibernate.cfg.Configuration;
import com.bean.classes.PathClass;
import com.bean.classes.StreetLight;
import Lpost Communication.com.traffic.communication.ResponseFromServer;
public class MainClass {
     public static void main(string [ ] args) {
          Configuration cfg=new Configuration( );
          cfg.configure("resourcefiles/hibernate.cfg.xml");
          SessionFactory sf = cfg.buildSessionFactory( );
          Session s=sf.openSession( );
          StreetLight lpostDetails=new StreetLight("Lpost_1", "Lane1");
          ResponseFromServer response=new ResponseFromServer( );
          response.getResponse(request, isFaulty, isAmbLightCondition,
          isTrafficCondition, status);
          response.getResponse(lpostDetails, 0, 1, 1, 0);
          PathClass finalobject = response.action(lpostDetails);
          s.save(finalobject);
          s.beginTransaction( ).commit( );
          s.close( );
          sf.close( );
          System.out.println("completed");
     }
}
```

Object JSONText out to string

The code in Java for adding a new sensor for traffic density is given below.

```
JSONObject obj=new JSONObject( );

    obj.put("name", "SensTrafficDensity");
```

Density is Integer

```
    obj.put("trafficDensitySensor", new Integer(100)); //Sensed Traffic
```

Boolean for new Sensor Addition

```
    obj.put ("is_newSensorAdded", new Boolean(true)); //Returns

    StringWriter out = new StringWriter( );

        obj.writeJSONString(out);

    String jsonText = out.toString( );

    System.out.print(jsonText);
```

Result : {"Name" : "SensTrafficDensity1" : "TrafficDensity", 10"
"numberOfSensors" : 1000000, "newSensorAdded" : null, }

Lamppost WSN StLight Class

(Following is the code in Java for class StLight with a lamppost ID) laneID.status,

```
package com.bean.classes;

public class StLight {

public StLight(String lpostId, String laneId) {

    this.lpostId = lpostId;

    this.laneId = laneId;

    }

    private int status; // status of functioning, traffic presence and ambient
```

light at a lamppost

```
    private String lpostId

    private String laneId;

    private int isFaulty; // Street Light if faulty boolean value is 1 mean (true)

    private int isLightCondition; // sensor value, if ambient light conditions
```

OK then value is 1 //(true)

```
    private int isTrafficCondition; // Traffic present the is 1 (true)

    public int getStatus( ) {

    return status;
```

```
    }

    public void setStatus(int status) {

    this.status = status;

    }

    public String getLpostId( ) {

    return lpostId;

    }

    public void setLpostId(String lpostId) {

    this.lpostId = lpostId;

    }

    public String getLaneId( ) {

    return laneId;

    }

    public void setLaneId(String laneId) {

    this.laneId = laneId;

    }

    public int getIsFaulty( ) {

    return isFaulty;

    }

    public void setIsFaulty(int isFaulty) {

    this.isFaulty = isFaulty;

    }

    public int getIsLightCondition( ) {

    return isLightCondition;

    }

    public void setIsLightCondition (int isLightCondition) {

    this.isLightCondition = isLightCondition;

    }

    public int getIsTrafficCondition( ) {

    return isTrafficCondition;

    }

    public void setIsTrafficCondition(int isTrafficCondition) {
```

```java
        this.isTrafficCondition = isTrafficCondition;
    }
}

[Class for Lamppost Communication of traffic information with a Coordinator
(Server) at an instance]
package LpostCommunication.com.traffic.communication;
import java.text.SimpleDateFormat;
import java.util.Date;
import com.bean.classes.PathClass;
import com.bean.classes.StreetLight;
public class ResponseFromServer {
    public StreetLight getResponse (StreetLight request, int isFaulty, int
isLightCondition, int isTrafficCondition, int status) {
        request.setIsFaulty(isFaulty);
        request.setIsLightCondition (isLightCondition);
        request.setIsTrafficCondition(isTrafficCondition);
        request.setStatus(status);
        return request;
    }

    public PathClass action(StreetLight response) {
        if(response.getIsLightCondition() == 1 &&
            && response.getIsTrafficCondition() == 1
            response.getIsFaulty() == 1
            System.out.println ("Light Condition :" + response.getIsLightCondition
( )+"Faulty"+response.getIsFaulty()+"Traffic Condition" +
response.getIsTrafficCondition( ));
        } else if (response. getIsLightCondition() == 1 && response.getIsFaulty() == 1
            && response.getIsTrafficCondition() == 0) {
            System.out.println ("Light status :" + response.
getIsLightCondition() + "Faulty" + response.getIsFaulty() + "Trafficstatus"
            + response.getIsTrafficCondition() ;
        } else if (response.getIsLightCondition () == 1 && response.
```

```java
getIsFaulty() == 0
            && response.getIsTrafficCondition() == 1) {
            System.out.println("Light status :" + response.getIsFaulty( )+"Faulty"
            + response.getIsLightCondition( ));
        } else if (response.getIsTrafficCondition( ) == 0) {
getIsFaulty() == 0
            && response.getIsTrafficCondition( ) == 1 && response.
getIsLightCondition() + "Faulty"+response.getIsFaulty() + "Traffic status"
            + response.getIsTrafficCondition( );
        } else if (response.getIsLightCondition ( ) == 0 & & response.
getIsTrafficCondition( ) == 1
            && response.getIsTrafficCondition( ) == 1) {
            System.out.println ("Light status:" + response.
getIsLightCondition( ) + "Faulty"+response.getIsFaulty( ) + "Traffic status"
            + response.getIsTrafficCondition( );
        } else if (response.getIsLightCondition ( ) == 0 & & response.
getIsFaulty( ) == 1
            && response.getIsTrafficCondition( ) == 0) {
            System.out.println ("Light status :" + response.
getIsLightCondition( ) + "Faulty"
            + response.getIsFaulty( ) + "Traffic status" + response.
getIsTrafficCondition( );
        } else if (response.getIsLightCondition ( ) == 0 && response.
getIsFaulty( ) == 1
```

```java
getIsLightCondition( ) + "Faulty" + response.getIsFaulty( ) + "Traffic status"
            + response.getIsTrafficCondition( );
        } else if (response.getIsTrafficCondition( ) == 0) {
getIsFaulty( ) == 0
            & & response.getIsTrafficCondition( ) == 0
            System.out.println("Light status :" + response.
getIsLightCondition( ) + "Faulty" + response.getIsFaulty( )+"Traffic status"
            + response.getIsTrafficCondition( );
        } else if (response.getIsTrafficCondition( ) == 1) {
            System.out.println ("Light status :" + response.
getIsLightCondition( ) + "Faulty"
            } else if (response.getIsTrafficCondition( ) == 0 & & response.
getIsFaulty( ) == 0
            System.out.println("Light status :" + response.
getIsLightCondition( ) + "Faulty" + response.getIsFaulty( )+"Traffic status"
            + response.getIsTrafficCondition( );
        } else {
```

```
System.out.println ("Light status :" + response.
getIsLightCondition() + "Faulty"
+ response.getIsFaulty() + "Traffic status" + response.
getIsTrafficCondition() );

}

return this.InsertOperations (response);

}
```

Class for Received Lampost Info insertion into Database at the Coordinator

```
public PathClass InsertOperations (StreetLight map)
{

PathClass Object=new PathClass();
SimpleDateFormat format=new SimpleDateFormat();
SimpleDateFormat tformat = new SimpleDateFormat("DD/MM/YYYY");
SimpleDateFormat tformat = new SimpleDateFormat ("HH:MM:SS");
Date date=new Date();
object.setAvgOnPeriod (10.0);
object.setAvgTrafficDensity (10.0);
object.setAvgTrafficPresence (10.0);
object.setDate(format.format (date));
object.setFaulty(map.getIsFaulty());
object.setLightFunctionality ("" +map.getIsLightCondition());
object.setRoadPathId (map.getLaneId());
object.setStatus ("" +map.getStatus());
object.setSubGroup("Left/right");
object.setTime (tformat.format (date));
object.setTimeSlotNumber (1);
return object;

}
}
```

Q.81. Explain IoT based smart farming in detail.

Ans. Smart farming based agriculture IoT stick is considered as IoT gadget focusing on live monitoring of environmental data in terms of temperature, moisture and other types depending on the sensors integrated

with it. Agricultural IoT stick provides the concept of "Plug & Sense" in which farmers can directly implement smart farming by as such putting the stick on the field and getting live data feeds on various devices like smart phones, tablets etc. and the data generated via sensors can be easily shared and viewed by agriculture consultants anywhere remotely via cloud computing technology integration. IoT stick also enables analysis of various sorts of data via Big data analytics from time to time.
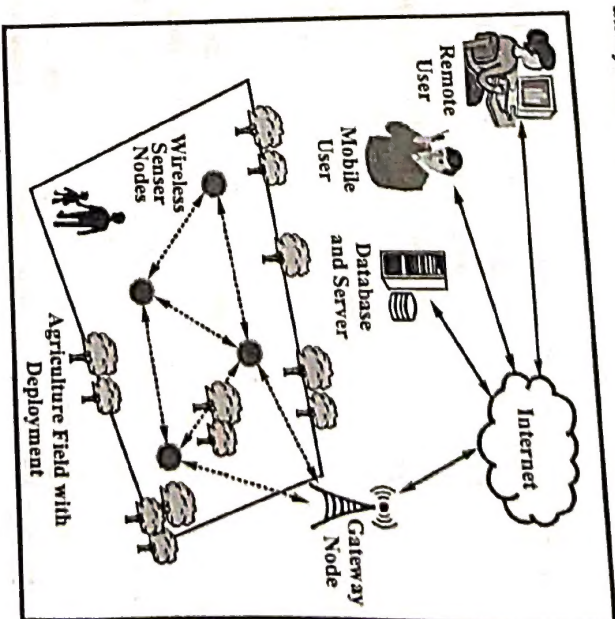

Fig. 5.34 Smart Agriculture

Various components i.e., modules and sensors are used for smart IoT agricultural stick development. These are discussed below –

(i) Arduino Mega 2560 – This module is designed for developing Arduino based robots and doing 3D printing technology based research. Arduino Mega 2560 is based on ATmega2560 and consists of 54 digital input/output pins, 16 analog inputs, 4 UART (Universal Asynchronous Receiver and Transmitter). It can simply connect to PC via USB port.

(ii) BreadBoard BB400 – This module is a solderless breadboard with 400 connection tie points i.e. 400 Wire insertion points. BB400 has a 300 tie-point IC-circuit area plus four 25-tie point power rails. Housing is made of White ABS plastic, with a printed numbers and letters of rows and columns.

Technical Specifications of BreadBoard BB400 are 36 volts, 2 Amps, 400 tie points, 50000 insertions.

(iii) ESP 8266 – This Wi-Fi module is SOC with TCP/IP protocol stack integrated which facilitates any microcontroller to access Wi-Fi network. ESP8266 module is cost effective module and supports APSD for VOIP applications and bluetooth co-existence interfaces. Technical specifications of ESP8266, 802.11b/g/n: Wi-Fi direct, 1 MB flash memory, SDIO 1.1/2.0, SPI, UART, Standby Power Consumption of < 1.0 mW.

(iv) Soil Moisture Sensor – It detects soil moisture. The sensor has both analog and digital output input and operates according to the principle of system. When the ground is dry, the electricity stops flowing and acts as an open circuit. If the ground is wet, the current passes and the circuit is short and the output is zero. Sensor information is indicated by levels. It is corrosion resistant so the sensor has a long time to handle the cost of the farmer at minimal cost.

(v) Temperature and Humidity Sensor – It is used to measure temperature and humidity. This system displays information about how well it worked. If the threshold is exceeded, the LED stars flashing and the values are immediately displayed on the web page and the farmer can check them.

(vi) Acoustic Sensors – These offers various uses in managing the farms, which includes cultivation of soil, weeding and harvesting. Major advantage of using this sensor is its quick response and low cost, mainly while making an allowance for portable equipment. These type of sensors work by measuring the alterations in the noise. These sensors are mainly used for monitoring of pests and its detection, variety of seeds are classified as well by using these sensors.

(vii) Optical Sensors – Optical sensors use a phenomena called light reflectance, which measures the organic substances in soil, moisture, minerals, colour and composition, etc. Ability of soil to reflect light depends upon the various parts of the electromagnetic spectrum are tested by these sensors. Variation in the soil density are indicated by the alteration occurred in the reflection of waves.

The hierarchy of key applications, services and wireless sensors being used for smart farming application is shown in fig. 5.35.
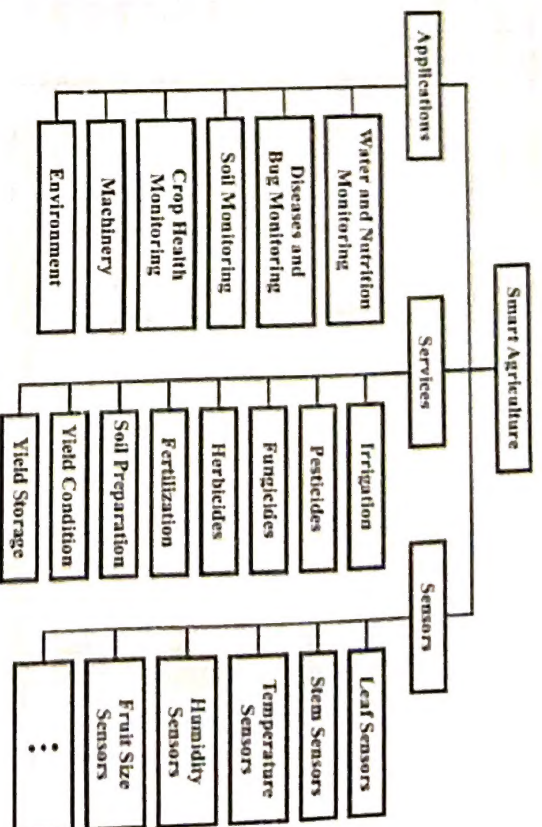
Applications — Smart Agriculture

Services: Water and Nutrition Monitoring, Diseases and Bug Monitoring, Soil Monitoring, Crop Health Monitoring, Machinery, Environment

Irrigation, Fungicides, Pesticides, Herbicides, Fertilization, Soil Preparation, Yield Condition, Yield Storage

Sensor: Leaf Sensors, Stem Sensors, Temperature Sensors, Humidity Sensors, Fruit Size Sensors

Fig. 5.35 General Hierarchy of Applications, Services and Sensors for Smart Farming

Q.82. What are the technologies used in smart farming?

Ans. Technology used for smart farming are vehicles which are equipped by GPS trackers and wireless sensors which are deployed in the fields to check the moisture and other parameters. Robots helps farmers by watering, planting, picking and fertilizing. Acoustic sensors are used for monitoring and control of pests and differentiating the variety of seeds.

(i) Database – All the crop data is saved in the database securely, whenever required, user can access the data.

Cellular communication is used to get the updates from the field, depends on the bandwidth and other factors user can receive updates. Cost of this communication is low. Using satellites for transmitting the data is another way of communication, however, it is costly for small farmers.

(ii) ZigBee – It is used in the small range to communicate. Depending on the applications it uses network topologies like star. Depends on the distance from farm land to the user GSM or Bluetooth can be used.

(iii) Bluetooth – This communication is wireless and operates at a shorter range from device to device. Due to its benefits such as low power consumption and low cost it is most preferable option to use. It can

communicate data like various parameter readings. It is available in any smart phone, unlike WiFi, which requires LAN connection to operate. However, WiFi can be used in large farms to operate, because lot of devices are connected, so it is easy to operate. Depending on the size of the farm land technology changes.
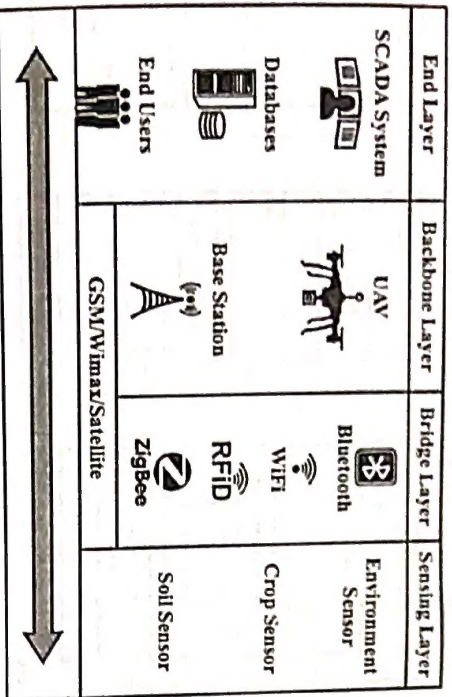
End Layer	Backbone Layer	Bridge Layer	Sensing Layer
SCADA System	UAV	Bluetooth / WiFi	Environment Sensor
Databases	Base Station	RFID / ZigBee	Crop Sensor
End Users	GSM/Wimax/Satellite		Soil Sensor

Fig. 5.36 Communication in Smart Farming

Q.83. Explain the following IoT applications –
(i) Soil sampling and mapping
(ii) Irrigation
(iii) Fertilizers
(iv) Greenhouse farming.

Ans. **(i) Soil Sampling and Mapping** – The purpose of this application is examination of soil which is very crucial to extract field specific information. It is useful while making decisions at certain stages. Main objective of analysing soil is to check the nutrient status of the farm land, so particular measures are taken depends on the deficiency of the field. Some factors that's helps us to analyse soil nutrients includes type of soil, history of crop, fertilizers, irrigation level, etc. There are many manufacturers who are providing sensors and tools which helps in soil testing, these kits assist farmers to know the quality of the soil. Depends on the data provided, remedies can be taken to increase the growth of the crop.

Drought is one of the main concern for the limitation of crop yield productivity. This issue can be resolved by using remote sensors are used to get the soil moisture data frequently to their mobile as a SMS alert and e-mail alert or check in the webpage. Depending on the data received we can allow

the water pump to flow, when it reaches the certain range we will get the alert, so that we can stop the water.

(ii) Irrigation – There are several irrigation methods, such as drip irrigation and sprinkler irrigation, that could handle the issues like water wastage, which were also found in traditional methods like flood irrigation and furrow irrigation. Water shortage can affect the crops quality and quantity and leads to the deficiency of the soil nutrients and develops infections which are harmful to the plants. Estimating water needed for the crops is very hard, it involves factors such as method of irrigation, type of crop, soil type and moisture level in the soil. Taking this into consideration, wireless sensors are used by the soil and moisture control system it helps the crop to use the water optimally and results in better crop health. IoT techniques are used to better the crop efficiency, some of the techniques are crop water stress index irrigation management. Wireless field sensors are installed to measure the conditions and transmit the data. (There are other data such as weather information and satellite imaging is used to calculate crop water stress index (CWSI), this increases the efficiency of water usage.

(iii) Fertilizers – Now-a-days smart farming uses fertilization which helps in the estimation of nutrients dosage requisition, due to this, negative effect on the environment will be reduced. With the use of smart agriculture dosage of nutrients is estimated, ultimately reduces their negative impact on the environment. The requirements for fertilization are type of soil, crop variety and absorption capacity, yield, fertility type and weather, etc. Satellite images help us to know the status of crop nutrients, we can easily asses the level of soil nutrients, these can further increase the efficiency of the fertilizer. Accuracy of GPS, autonomous vehicles are advantageous to smart fertilization. IoT based devices, smart robots and drones are used to detect the pests and control them by sprinkling pesticides.

(iv) Greenhouse Farming – It is considered as one of the method of smart agriculture. However, growing crops in the controlled environment is not a new idea, as it has roots from 19th century. But this kind of setup is used in countries which experience drastic changes in climatic conditions. The day's crops grow in indoors are not much affected by varied environmental conditions and also we can grow any kind of crops. In these times the support of sensors and other devices are important for communication. There are some factors which can influence the crops production such as ventilation system, monitoring and wind control, etc. In modern greenhouse farming, the monitoring of environment is the major task. The monitoring of environment is the major task. Where we have to check the different measuring points which control the indoor climate.

Q.84. What are the benefits and drawback of IoT in agriculture ?

Ans. The benefits of IoT in agriculture are as follows —

(i) IoT enables easy collection and management of tons of data collected from sensors and with integration of cloud computing services like agriculture fields maps, cloud storage etc., data can be accessed live from anywhere and everywhere enabling live monitoring and end to end connectivity among all the parties concerned.

(ii) With IoT productions costs can be reduced to a remarkable level which will in turn increase profitability and sustainability.

(iii) IoT is regarded as key component for smart farming as with accurate sensors and smart equipment's, farmers can increase the food production by 70% till year 2050 as depicted by experts.

(iv) With IoT, various factors would also lead to the protection of environment.

(v) With IoT, efficiency level would be increased in terms of usage of soil, water, fertilizers, pesticides etc.

Some drawback of IoT in agriculture are as follows --

(i) There could be a wrong analysis of weather conditions.

(ii) If there are faulty data processing equipment or sensors, then it will lead to a situation where the decisions are taken wrong.

(iii) Devices are to be altered according to the farmers, it will involve equipment which will be expensive.

●●●